Identifying Problems in Onboarding Design for Expertise-Centric Citizen Science Games

A Dissertation Presented
by

Joshua Aaron Miller

to

The Department of Khoury College of Computer Sciences

in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

in

Computer Science

Northeastern University
Boston, Massachusetts

April 2023

DocuSign Envelope ID: 36C4C5AE-97C1-40B4-B1C2-C33AD8BAC489

Northeastern University
Khoury College of
Computer Sciences

PhD Thesis Approval

Thesis Title: Identifying Problems in Onboarding Design for Expertise-Centric Citizen Science Games

Author: Joshua Aaron Miller

PhD Program: X Computer Science ____ Cybersecurity ____ Personal Health Informatics

PhD Thesis Approval to complete all degree requirements for the above PhD program.

DocuSigned by: Seth Cooper 8310CD3FBA654AE...	5/24/2023
Thesis Advisor	*Date*
DocuSigned by: Magy Seif El-Nasr 3910D4E36A0548C...	5/11/2023
Thesis Reader	*Date*
DocuSigned by: Casper Harteveld 349D5FD806184EA...	5/9/2023
Thesis Reader	*Date*
DocuSigned by: Sebastian Deterding 7CFCAC224D3E480...	5/24/2023
Thesis Reader	*Date*
Thesis Reader	*Date*

KHOURY COLLEGE APPROVAL:

Amal Taak	5/25/2023
Associate Dean for Graduate Programs	*Date*

COPY RECEIVED BY GRADUATE STUDENT SERVICES:

Laura E Ad	26 May 2023
Recipient's Signature	*Date*

Distribution: Once completed, this form should be attached as page 2, immediately following the title page of the dissertation document. An electronic version of the document can then be uploaded to the Northeastern University-UMI Website.

For confused learners everywhere.

Contents

List of Figures x

List of Tables xii

List of Acronyms xiii

Acknowledgments xv

Author's Preface xvii
0.1 How to Read This Dissertation . . . xvii
0.2 Other Works . . . xvii

Abstract of the Dissertation xix

I Introduction 1

1 The Hardest Games to Learn 2
1.1 Problem Statement . . . 6
1.2 Approach . . . 7
1.3 Research Areas . . . 9
1.4 Significance and Contributions . . . 11
1.5 Thesis Statement . . . 12
1.6 Thesis Overview . . . 12

2 What We Know 15
2.1 Citizen Science . . . 15
2.1.1 Citizen Science Games . . . 16
2.1.2 Motivation and Engagement . . . 17
2.1.3 Accessibility and Inclusion . . . 18
2.2 Related Terms . . . 20
2.3 Expertise-Centric Citizen Science Games . . . 21
2.3.1 Scientific Contributions . . . 27
2.4 Learning in Games . . . 28

2.4.1 Expertise . . . 28
2.4.2 Onboarding Design . . . 30
2.5 Conclusion . . . 32

3 A Naive Approach **33**
3.1 Introduction . . . 33
3.2 Background . . . 35
3.2.1 Self Determination Theory . . . 35
3.2.2 Cognitive Load Theory . . . 35
3.2.3 Usage of SDT and CLT . . . 37
3.2.4 Personalized Game Design . . . 38
3.3 Methodology . . . 39
3.3.1 Theory Adaptation . . . 39
3.4 Experimental Evaluation . . . 44
3.4.1 Experiment Design . . . 44
3.4.2 Analysis Methods . . . 45
3.4.3 Results . . . 46
3.5 Discussion . . . 47
3.5.1 Limitations . . . 49
3.6 Conclusion . . . 50
3.6.1 Takeaways . . . 51

II Identifying Problems **52**

4 The State of Play **53**
4.1 Player Study . . . 53
4.1.1 Methods . . . 54
4.1.2 Participant backgrounds . . . 55
4.1.3 Update preferences . . . 55
4.1.4 Tutorial experiences . . . 57
4.1.5 Game difficulty . . . 58
4.1.6 Open-ended game feedback . . . 59
4.2 Stakeholders Study . . . 62
4.2.1 Methods . . . 64
4.2.2 Overarching Themes . . . 66
4.2.3 Takeaway Recommendations . . . 79
4.2.4 Discussion . . . 83
4.3 Conclusion . . . 85
4.3.1 Takeaways . . . 86

5 Barriers to Expertise **87**
5.1 Background . . . 88
5.2 Methods . . . 92
5.2.1 Participants and Protocol . . . 92

5.2.2 Analysis . . . 93
5.3 Results . . . 95
5.3.1 The Path to Expertise is Social and Exploratory . . . 95
5.3.2 Missing Instruction (The Instructional Barriers) . . . 102
5.3.3 Missing Polish (The Game Barriers) . . . 107
5.3.4 Missing Communication (The Sociocultural Barriers) . . . 109
5.4 Discussion . . . 116
5.4.1 Limitations, Implications, and Future Work . . . 124
5.5 Conclusion . . . 125
5.5.1 Takeaways . . . 125

III Finding Solutions **126**

6 Tutorial Design Patterns **127**
6.1 Background . . . 128
6.1.1 Onboarding Design . . . 128
6.1.2 Close Reading Methodology . . . 129
6.2 Methods . . . 130
6.2.1 Game Selection . . . 130
6.2.2 Protocol . . . 131
6.3 Results . . . 133
6.3.1 Successful Games Pace Learning and Check Understanding . . . 133
6.3.2 Successful Games Set Expectations . . . 136
6.3.3 Polish Means Playable Not Pretty . . . 138
6.4 Discussion . . . 140
6.4.1 Limitations and Future Work . . . 144
6.5 Conclusion . . . 144
6.5.1 Takeaways . . . 145

7 Instructional Design of ECCSG Skills **147**
7.1 How do Players and Developers of CSGs Conceptualize Skill Chains? . . . 150
7.1.1 Background . . . 150
7.1.2 Methods . . . 155
7.1.3 Results . . . 160
7.1.4 Themes . . . 160
7.1.5 Discussion . . . 170
7.1.6 Conclusion . . . 180
7.2 A Skill-Based Cognitive Task Analysis of Foldit . . . 180
7.2.1 Four-Component Instructional Design . . . 181
7.2.2 Methods . . . 182
7.2.3 Results and Discussion . . . 183
7.3 Conclusion . . . 187
7.3.1 Takeaways . . . 187

8 Complex Solutions for Complex Learning **189**
8.1 Background . . . 190
8.2 Designing a New Onboarding . . . 190
8.2.1 Player Background Matters . . . 191
8.2.2 The Giant's Staircase . . . 194
8.2.3 Communicate, Communicate, Communicate . . . 198
8.2.4 Polish For Playability . . . 199
8.2.5 The Big Picture . . . 200
8.2.6 We Play in a Community . . . 201
8.2.7 Reveal, Review, Repeat . . . 202
8.2.8 Theory-Driven Features for Guiding Cognition . . . 203
8.2.9 Design Reflections . . . 204
8.3 Iterative Playtesting . . . 206
8.3.1 Methods . . . 206
8.3.2 Results . . . 207
8.3.3 Discussion . . . 208
8.4 Phase 1 Empirical Testing . . . 208
8.4.1 Methods . . . 209
8.4.2 Results . . . 211
8.4.3 Discussion . . . 214
8.5 Phase 2 Empirical Testing . . . 214
8.5.1 Methods . . . 214
8.5.2 Results . . . 215
8.5.3 Discussion . . . 216
8.6 Phase 3 Empirical Testing . . . 216
8.6.1 Methods . . . 216
8.6.2 Results . . . 218
8.6.3 Discussion . . . 222
8.7 Conclusion . . . 227
8.7.1 Takeaways . . . 228

9 Conclusion **229**

IV Appendix **239**

A Games Studied **240**
A.1 *ARTigo* . . . 240
A.2 *Eterna* . . . 240
A.3 *Eyewire* . . . 240
A.4 *Foldit* . . . 241
A.5 *Forgotten Island* . . . 242
A.6 *Happy Match* . . . 242
A.7 *Living Links* . . . 242
A.8 *Mozak* . . . 242

A.9 *Phylo* . . . 242
A.10 *Quantum Moves 2* . . . 242
A.11 *Questagame* . . . 243
A.12 *Reverse the Odds* . . . 243
A.13 *Skill Lab: Science Detective* . . . 243
A.14 *Stall Catchers* . . . 243

B CSG Player Experience Survey **244**
B.1 Questionnaire . . . 244
B.1.1 Participant backgrounds . . . 244
B.1.2 Update preferences . . . 244
B.1.3 Tutorial experiences . . . 245
B.1.4 Game difficulty . . . 245
B.1.5 Open-ended game feedback . . . 245
B.2 Results . . . 246
B.2.1 Participant backgrounds . . . 246
B.2.2 Update preferences . . . 246
B.2.3 Tutorial experiences . . . 247
B.2.4 Game difficulty . . . 247
B.2.5 Open-ended game feedback . . . 248
B.2.6 Overall . . . 249

C CSG Stakeholders Study Details **250**
C.1 Ethnographic Study Methods . . . 250
C.2 Results . . . 251
C.2.1 Participants . . . 253
C.2.2 Project Leads . . . 256
C.2.3 Professional Scientists . . . 257
C.2.4 Other Team Members . . . 259
C.2.5 Educators and Students . . . 264
C.2.6 Other Groups . . . 267

D Code Descriptions **268**

E Experiential Prompts **272**

F Summary of Games and Experiences **273**
F.1 7 Billion Humans . . . 273
F.2 Air Forte . . . 273
F.3 Apetopia . . . 274
F.4 Baba Is You . . . 274
F.5 Boson X . . . 275
F.6 Breaking Good . . . 275
F.7 Cancer Crusade . . . 275
F.8 ChromaGun . . . 275

F.9 Colony B . . . 276
F.10 Ding Dong XL . . . 276
F.11 Don't Escape: 4 Days to Survive . . . 276
F.12 EcoBuilder . . . 277
F.13 Eterna . . . 277
F.14 Eyewire . . . 277
F.15 Foldit . . . 278
F.16 FTL: Faster Than Light . . . 278
F.17 Game Dev Tycoon . . . 278
F.18 Gunpoint . . . 279
F.19 Ikaruga . . . 279
F.20 Influent . . . 279
F.21 Kerbal Space Program . . . 280
F.22 Lazy Galaxy . . . 280
F.23 Learn Japanese to Survive! Kanji Combat . . . 281
F.24 Lightmatter . . . 281
F.25 Luxor Evolved . . . 282
F.26 Lyne . . . 282
F.27 Mini Metro . . . 282
F.28 Mozak . . . 282
F.29 Niche, a genetics survival game . . . 283
F.30 Odyssey — The Story of Science . . . 283
F.31 Orbt XL . . . 284
F.32 Opus Magnum . . . 284
F.33 Papers, Please . . . 284
F.34 PC Building Simulator . . . 285
F.35 Phylo . . . 285
F.36 Plague Inc: Evolved . . . 286
F.37 Poly Bridge . . . 286
F.38 President for a Day — Floodings . . . 286
F.39 Project Hospital . . . 287
F.40 Quantum Minds . . . 287
F.41 Quantum Moves 2 . . . 287
F.42 Skill Lab: Science Detective . . . 288
F.43 Sokobond . . . 288
F.44 Starbound . . . 289
F.45 Super Hexagon . . . 289
F.46 The Room Three . . . 289
F.47 TIS-100 . . . 290
F.48 TowerFall Ascension . . . 290
F.49 Tyto Ecology . . . 290
F.50 Unheard — Voices of Crime . . . 291
F.51 Veritas . . . 291
F.52 War Solution — Casual Math Game . . . 291

F.53 while True: learn() . . . 292
F.54 Zoombinis . . . 292
F.55 Zup! 2 . . . 293

G Design Documentation **294**
G.1 Learning Tasks . . . 294
G.2 Design Principles . . . 297
G.3 Template for Lesson Design . . . 299
G.3.1 Lesson Objective . . . 299
G.3.2 Reason for Lesson . . . 299
G.3.3 Overview of Lesson . . . 299
G.3.4 Necessary concepts and processes . . . 300
G.3.5 Demonstration . . . 300
G.3.6 Practice with feedback . . . 300
G.3.7 Example . . . 300

H Final Foldit Re-Design **302**
H.1 World 1: Stabilize . . . 302
H.2 World 2: Fold . . . 303
H.3 World 3: Design . . . 304

I Foldit Empirical Testing Details **305**
I.1 Phase 1 Quiz . . . 305
I.2 Phase 3 Quiz . . . 308

Bibliography **311**

List of Figures

1.1 Player retention in the *Foldit* tutorial . . . 5
1.2 Overview of chapters . . . 13

3.1 Experimental features in *Foldit* . . . 41
3.2 Branching Levels . . . 41
3.3 Level re-completions by condition . . . 46
3.4 Prior domain knowledge and game expertise affect engagement . . . 48

4.1 Rankings of update preferences . . . 56
4.2 *Foldit* tutorial difficulty . . . 57
4.3 Diagram of typical channels of communication and information dissemination . . . 76

5.1 The path to expertise, visual summary . . . 95

6.1 Game camps versus pacing features . . . 134
6.2 Educational value . . . 137
6.3 Game camps versus polish features . . . 139

7.1 Example skill chain . . . 156
7.2 Examples of the process of playing . . . 167
7.3 Examples of equating the tutorial with onboarding . . . 167
7.4 Final skill diagram . . . 183
7.5 Sample of the gold standard procedure . . . 184
7.6 Excerpt of the tutorial design plan . . . 185

8.1 Campaign level selection screen . . . 192
8.2 The giant's staircase . . . 195
8.3 FAQ panel . . . 195
8.4 Examples of explanatory images . . . 197
8.5 *Foldit*'s main (mode selection) menu . . . 203
8.6 Participant reports about the skills learned in *Foldit* . . . 211
8.7 Participant reports of perceived difficulty over time . . . 212
8.8 Highest scoring solution in Phase 1 . . . 213
8.9 Scatter plot of Phase 3 participants' performance . . . 219

9.1 The road to expertise 234

A.1 Screenshots of some games studied 241

List of Tables

2.1 Comparison of ECCSGs and related concepts 23
2.2 Typology of citizen science input/output structures 28

3.1 Summary of main results 45

4.1 Summary of themes across multiple stakeholder groups. 67
4.2 List of Takeaway recommendations 80

5.1 Summary of findings and recommendations 122

6.1 Games selected 132

7.1 Data summary 156
7.2 Label categories 160
7.3 Comparisons of skill models 161

8.1 Summary of Development 192
8.2 Descriptive statistics for each condition 220
8.3 Participant performance 220
8.4 Performance results for high-scoring participants 221

9.1 Actionable Guidelines 232

B.1 Participant expertise data 247

C.1 Summary of stakeholder groups and needs/challenges 253

List of Acronyms

4C/ID Four-Component Instructional Design. A framework for instructional design by van Merriënboer which considers instruction in four components: learning tasks, supportive information, procedural information, and part-task practice [540, 539, 542, 538, 536, 537].

AI artificial intelligence. A field of computer science regarding the machine processing of information.

ARG alternate reality game. A transmedia game or story that uses the real world as a problem-solving platform, often requiring a crowd of players working together to solve its mysteries [264].

CLT Cognitive Load Theory. A theoretical framework of cognitive resources that states cognitive efforts are spent on intrinsic load, extraneous load, and germane processing [500, 501, 192].

CSG citizen science game. A game where players make real scientific contributions.

CTA Cognitive Task Analysis. A framework of methods for analyzing how cognitive tasks are performed [103].

ECCSG expertise-centric citizen science game. A CSG that requires expertise to make a contribution.

GBL game-based learning. The use of games or gamification for applications of learning.

GEL Guided Experiential Learning. A theory of instructional design which emphasizes guidance and authentic practice.

GUR games-user research. A subset of HCI which focuses on how users interact with games.

GWAP game with a purpose. A synonym for human computation games (HCGs) [550].

HCG human computation game. Similar to citizen science games (CSGs) in form and function (but not necessarily to solve scientific problems), these are any games which employ human intelligence for real-world problem solving [381].

HCI human-computer interaction. The field of how users interact with technology.

IMI Intrinsic Motivation Inventory. A validated measure of motivation grounded in Self-Determination Theory [409, 321].

ML machine learning. A branch of artificial intelligence which uses data as training material to imitate human learning, e.g., for classification of new data.

MLC Motivation-Learning-Creativity. A model of citizen science motivation by Jennett et al. [241].

MMORPG massively multiplayer online roleplaying game, such as *World of Warcraft* by Blizzard Entertainment and *RuneScape* by Jagex.

QCA qualitative content analysis. A qualitative analysis using a codebook to count instances of content, see [198, 312].

QM *Quantum Moves*. An ECCSG, the prequel to *Quantum Moves 2* (QM2).

SBCTA Skill-Based Cognitive Task Analysis. A CTA with an emphasis on skill modeling, see [467].

SDG scientific discovery game. An HCG specifically for scientific problems [97].

SDT Self-Determination Theory. A theoretical framework for motivation which states that intrinsic motivation comes from autonomy, competence, and relatedness [122, 123].

UI user interface. A term in HCI and GUR for the interface between a user and a technology. Most often, this is a digital screen. In games specifically, the UI refers to the elements of the screen which the player directly interacts with or receives information from, as opposed to elements within the game world. For example, a health bar, a button prompt, and a pause menu are all UI elements while the player character, enemies, and game world are typically not considered UI elements. When paired with UX (as in, “UI/UX”), it refers to the study and practice of developing user interfaces and user experiences.

UX user experience. A term in HCI and GUR for the user’s / player’s experience, or more generally the study of user experiences.

Acknowledgments

No dissertation is written alone. This work would not have been possible without the support of my friends, family, and colleagues, and I am eternally grateful for all of them.

First, immeasurable gratitude goes to my adviser, Dr. Seth Cooper. Over the last six years, Seth has gone above and beyond as an adviser and mentor. He funded my work, challenged my thinking, and advocated for my well-being. Like a good tutorial, he held my hand when I needed him and let go when I was ready. But more than that, he supported me wholly as a person and a researcher, and I could not ask for a better adviser.

Similar thanks extend to my thesis committee, Dr. Magy Seif El-Nasr, Dr. Casper Harteveld, and Dr. Sebastian Deterding. Each of these brilliant scholars challenged me intellectually, being equal parts unabashedly critical and earnestly supportive. I am honored to stand on the shoulders of these giants.

I further acknowledge the funding I received to conduct this work. Parts of this work were supported by the National Institutes of Health grant 1UH2CA203780. Some of this material is based upon work supported by the National Science Foundation under grant no. 1652537. Other parts of my research as a PhD student were supported by National Institutes of Health grant UH2EB024407, a Northeastern University TIER1 grant, the University of California, Davis, the University of Notre Dame, and the fellowship funding support of Khoury College of Computer Sciences. Parts of this work were also supported by the National Science Foundation under grant no. 1629879.

Nothing I've written was done in isolation. The articles adapted for this work were co-authored by colleagues such as Britton Horn, Jonathan Romano, Amy Robinson Sterling, Uttkarsh Narayan, Matthew Hantsbarger, Anna Gander, Colin Milburn, Ranjodh Singh Dhaliwal, Katherine Buse, Aditya Ponnada, and Libuše Veprek, to name but a few of the incredible people I've worked with. Indeed, I would not have been able to accomplish anything with *Foldit* were it not for the wonderful developers and scientists who supported my research and answered every dumb question I asked them. Many, many thanks to Brian Koepnick, Jeff Flatten, Firas Khatib, Rocco Moretti, Scott Horowitz, Thomas Scott, Ian Haydon, and Ariana Cohn. My extended thanks to the larger RosettaCommons community, including Andrew Leaver-Fay and Jared Adolf-Bryfogle — who helped me get started in Rosetta's code — and Rhiju Das, lead on *Eterna*.

I am also deeply grateful to the many people who enabled my research, including every editor and anonymous reviewer and all of my participants. A special thanks to the *Foldit* players, who are enormously helpful and make up one of the nicest corners of the internet. I am especially thankful to the veteran players who helped me learn how to play and conversed with me in *Foldit*'s chat, including Susume, LociOiling, Skippysk8s, alcor29, Boots McGraw, Bruno Kestemont, cjddig, Formula350, HuubR, jeff101, NinjaGreg, nspc, and S0ckrates. May you all see the fruits of your citizen science efforts.

Thanks to globalization of academia, I have had the pleasure of connecting with educators and researchers from around the world, and I am grateful for the conversations, connections, and collaborations with Ioanna Iacovides, Elisa Mekler, Janet Rafner, Joseph Hesketh, Matthew Whitby, Jacob Sherson, Carlos Díaz, Chris Barney, Bob De Schutter, Celia Pearce, Chris Martens, Alexandra To, Joseph Brown, Hamna Aslam, Jessica Hammer, Aaron Trammell, Eddie Melcer, Gillian Smith, Kate Compton, Theresa Jean Tanenbaum, and many other amazing scholars.

I was also very fortunate to have been a part of the GUII lab at Northeastern University. To Elin Carstensdottir, my first mentor as a PhD student, thank you for taking us fledglings under your wing. Thank you Nathan Partlan, Abdelrahman Madkour, Chaima Jemmali, Dylan Schouten, Paola Rizzo, Sarah Bunian, Andy Bryant, Sabbir Ahmad, Colan Biemer, and Kutub Gandhi for the good times of camaraderie, the games, the whiteboard shenanigans, and the intellectual stimulation. An extended thanks to the entire Northeastern graduate student community for their company and solidarity — here is to hoping that all workers, graduate student or otherwise, receive better support from their employers.

On a more personal note, and especially given the global pandemic, I give sincere thanks to my friends who supported me through this journey with good company. You know who you are, and I will always be here for you.

To my mom, for her infinite love and support, thank you from the bottom of my heart. You have given me the most support of anyone — 28 years of it — and it is your love that empowers me in all that I do. You are the Aaron in Josh Aaron Miller, and I will honor you and cherish you always with every moment of my life and every publication that bears the Aaron name.

Lastly, but certainly not least, an immense and heartfelt thank you to my partners, lovingly “the waffles,” for your constant mental, physical, and emotional support. Together, we can do anything.

Author's Preface

0.1 How to Read This Dissertation

Hello! If you're reading this, you're probably either a game developer or a games academic (or both!) and you're interested in tutorial design and/or citizen science games. To try to save you some time, I've made sure this dissertation can be skimmed effectively in two ways:

- **The 10-minute read**: If you're just looking for a brief, blog-post-sized overview, jump straight to the conclusion (Chapter 9).
- **The committed skim**: If you're willing to spend 30–60 minutes getting the big picture and diving into the part that matters to you:

    1. Read Chapter 1, which describes the other chapters at the end of it
    2. For every chapter, skim them for the figures, then read the brief Takeaways section included at the back of every chapter (with the exception of Chapter 2 which is mostly academic background work)
    3. Read the conclusion (Chapter 9)
    4. If any chapter(s) interested you, go back to read their details as little or as much as you want.

You'll notice I haven't included a workflow that includes reading the entire dissertation from front to back; this is reserved for my thesis committee (sorry[1]) and anyone deeply interested in expertise-centric citizen science game (ECCSG) research.

To anyone reading this, thank you for your interest, and if you would like to follow up on anything, you can reach me at `josh@joshaaronmiller.com`.

0.2 Other Works

Although I have spent the last six years studying citizen science games and onboarding, not all of my work went into this dissertation. Citizen science games are but one type of game which requires expertise, and I am more interested in expertise and onboarding in games generally. Given that, I'd like to take a moment to describe some of the other works I've authored during my time as a PhD student:

[1] In the words of Benjamin Franklin, "I have already made this paper too long, for which I must crave pardon, not having now time to make it shorter" [167].

In *Roleplaying as a solution to the quarterbacking problem problem of cooperative and educational games* [329], I describe the quarterbacking or "alpha gamer" problem, why it's a problem for cooperative and educational games, and synthesize a framework of solutions to it, focusing especially on roleplaying as a somewhat novel approach. Along the same theme of roleplaying, in *The Player-Learner Experience: A Comparison of Game Masters and Pedagogical Practices* (to be published) I compare tabletop roleplaying Game Masters (GMs) to teachers, showing how their roles are similar. I identify a few principles of interaction design which I believe are used successfully by GMs and teachers, and I give practical descriptions for how GMs and teachers can implement these ideas for themselves.

In *Designing for Reflective Play: A Practical Toolkit* (to be published), my co-authors and I create a toolkit of design patterns for generating reflective play and how developers can use these patterns for encouraging players to reflect on their games. Also oriented toward commercial games, in *Case Studies in Game-Based Complex Learning* [331] I examine several popular gameful and gamified applications which include complex learning to draw insights about what makes successful complex learning in games.

Finally, based on *Foldit*, I've published several articles only tangentially related to the work of this dissertation. In *Wrapped in Story: The Affordances of Narrative for Citizen Science Games* [330], my co-authors and I measured the impact of a science fiction narrative implemented into *Foldit*. Similarly, *Introducing Foldit Education Mode* [335] describes the addition of an educational mode, and *Effects of Player-Level Matchmaking Methods in a Live Citizen Science Game* measures the impact of a practice mode (Dojo mode) using skill-based matchmaking methods to pair players dynamically with levels of appropriate difficulty. Lastly, in *Large-Scale Analysis of Visualization Options in a Citizen Science Game* [336], I analyzed differences between novice and expert *Foldit* players in how they use the wide array of view options available to them to solve the game's problems.

Alongside these academic articles, I've released three small games,[2] and I plan on continuing to create games which experiment with novel ideas and support complex learning using the tools I've studied throughout my academic career. I could write much more on the value of academics making games and developers experimenting, but you're reading this for the dissertation itself, so without further ado...

[2] https://joshaaronmiller.itch.io/

Abstract of the Dissertation

Identifying Problems in Onboarding Design
for Expertise-Centric Citizen Science Games

by

Joshua Aaron Miller
Doctor of Philosophy in Computer Science
Northeastern University, April 2023
Dr. Seth Cooper, Adviser

In the expertise-centric citizen science game (ECCSG) model of scientific research, researchers develop a game which enables citizen scientists to gain a complementary domain expertise that contributes new knowledge to the domain. However, the effectiveness of this model is reliant on the onboarding experience for training and retaining players. Yet, there are few leads on what most needs improving within the onboarding of citizen science games. Therefore, the purpose of this dissertation is to discover what areas of design can be most improved in the onboarding of ECCSGs through a design science approach. The driving research questions are: (1) How is ECCSG onboarding currently designed and perceived? (2) What design changes are practically effective in improving ECCSG onboarding? To answer these questions, I use a combination of Games User Research techniques such as surveys, interviews, and cognitive task analysis, in addition to "close play" and thematic analysis, in order to build a model of how players perceive and experience the tutorial and the skills they learn from it, extending previous research on player modeling and skill chains. Finally, I test the practical efficacy of the theories produced by implementing design insights into *Foldit*, a popular ECCSG. While a few features showed promise, such as signaling and dynamic advice, I conclude that the four major components of designing for expertise in ECCSGs are practice, community, simplicity, and explanation.

Part I

Introduction

Chapter 1

The Hardest Games to Learn

Imagine that there are video games where — just by playing them — you could contribute to developing a treatment for AIDS. Imagine that, on your phone, you could develop novel treatments for cancer while commuting to work. In fact, these games exist [340, 261]. This is the promise of citizen science games (CSGs). Play a game and help real scientists with real research [97]. Do they work? Sometimes!

From a task-based perspective [255], there are two kinds of citizen science. The first, data-centric projects, are projects where the users handle data for the scientists [255]. In *Galaxy Zoo*, for example, users are shown pictures of galaxies and need to classify them to survey our universe [415]. And on *iNaturalist*, you can upload pictures of local plants and animals so biologists can track biodiversity [365]. For these projects, the contribution is simply the power of the crowd to collect, create, and annotate. It's all about big data. These projects work reliably to answer data-driven questions, so long as the project can recruit and engage a wide enough audience [551].

The second kind of citizen science is expertise-centric [255]. Here, players don't work with data, they work with problems. Can we design a protein to bind to the novel coronavirus [99, 274]? Can we develop a better test for tuberculosis [438, 140]? Similar to alternate reality games (ARGs),[1] this approach is about bringing together a crowd to solve hard problems.

While the scientists are experts of their domain, players of expertise-centric citizen science games (ECCSGs)[2] are tasked with gaining a *complementary* expertise and teaching the scientists how to solve problems that they don't yet know how to solve [255].

[‡]Parts of this chapter were adapted from [332].

[1]A transmedia game about problem-solving in the real world, often requiring a crowd of players working together to solve its mysteries [264].

[2]While Keep [255] was the first to coin the term "expertise-centric" citizen science, he did not use the acronym ECCSG to refer to expertise-centric citizen science games as a category, and his definition was not as rigorous as I define it in Chapter 2.

ECCSGs represent an important new kind of gaming. The twenty-first century continues to show a need for effective mass cooperation and interpersonal efforts in a variety of ways. ECCSGs are a microcosm of interdisciplinary and complementary expertise that we can study to understand how to solve larger and more complex problems than ever before. By understanding expertise development in ECCSGs, we may be able to gain new insights into how to train learners in new domains for which little to no training materials have yet been developed.

But ECCSGs, indeed all CSGs, serve two other purposes as well. First, CSGs are a means of socially accessible science, "providing the public with access to important and challenging problems facing science and society" [523]. This includes not only being able to physically access opportunities to contribute to science, but cognitive access (understanding why they are able to contribute and how their contribution is important) and social access (making contributions a socially acceptable, and even encouraged, activity to engage in). Second, CSGs create real scientific advancement: empowering citizens to solve these challenging problems which could otherwise take decades of scientific effort — increasing the quality and quantity of scientific data for research in fields like protein and RNA design and virtual neuron reconstruction — the tasks of *Foldit*, *Eterna*, and *Eyewire* respectively [99, 438, 514].

In theory, ECCSGs could be a new means of scientific knowledge production — as important as recent breakthroughs in artificial intelligence (AI) and machine learning (ML). But in practice, they are niche novelties at the edges of gaming and citizen science alike. Why?

As expertise-centric games, they take months — or even years — of play to fully understand the domain enough to make meaningful contributions. For these projects to achieve their goals, they have to train their players in how to play: they have to bring their players to that level of expertise. In practice, this is very, very difficult. ECCSGs are, in my opinion, the hardest games in the world to learn.

ECCSGs are at a turning point, a watershed moment. The website `citizensciencegames.com` shows that CSGs spiked in popularity around 2012 and dropped off in the last five years. If they continue failing to engage and maintain an audience, we may see widespread distrust, splintering, or abandonment of the citizen science gaming model entirely.

There may be several reasons why ECCSGs remain niche, but I argue that the most important one to address right now is the challenge of onboarding. ECCSGs rely heavily on their onboarding experience.[3] For three reasons, I argue it is in fact the most important aspect of ECCSGs:

1. **Primacy** — The tutorial includes the first moments experienced by the player. This sets the foundation for expectations moving forward. If the first moments are enjoyable, the player can infer that the rest of the game will be similar. If they aren't, the player may not make it to the rest of the game anyway.

2. **Visibility** — Because the tutorial is the first player experience, it is also the one seen by the most players. As the game progresses, fewer and fewer players will be retained. This is especially significant in CSGs [450]. Therefore, it makes sense to first focus on the part of the game which all players will reach.

3. **Impact** — The tutorial sets the player's mental model of the game. If they are confused, frustrated, or lost in learning how to play, they will be discouraged from playing again. If, however, they feel (and *are*) competent, they are much more likely to continue playing and be effective at the game's tasks. The learnability of a game is a key factor in continued use according to the Technology Acceptance Model (TAM) [117, 196].

Indeed, the "first hour" of initial play has been shown to be critical for sustained engagement [77], and there is recent evidence that tutorials improve flow and continuous-use intentions in non-experts [379]. If the purpose of ECCSGs is to cultivate expertise, then any efforts spent enabling the player to gain that expertise (e.g., through good tutorialization and adequate learning resources) will be contributing efficiently to the project's goals.

What are the objectives in designing an effective onboarding experience? Intuitively, there are two: to teach and to motivate. Teaching involves not only training the player in the game's mechanics, but also acclimatizing and acculturating the player to the expectations of the game and its community, as well as managing the stress of engaging with something new via comfort and respect [549]. Crumlish and Malone [108] describe this as *accommodation*, *assimilation*, and *acceleration*. Accommodation is providing the requisite tools; assimilation is bringing the user into the culture; and acceleration is engaging with the full feature set quickly and efficiently.

Motivation includes providing intrigue and excitement which prompts continued play and investment [549, 77]. From previous research (e.g., [114, 241]) we know fairly clearly what moti-

[3] A brief tangent for jargon: *onboarding* is the general process of tutorialization, teaching, and assimilation. Most often when we talk about onboarding, we refer to the *tutorial*, which is a designed module for the bulk of the onboarding process, typically introduced at the beginning of the game. Tutorials are definitionally a subset of the onboarding experience, although sometimes they are the only onboarding a game offers.

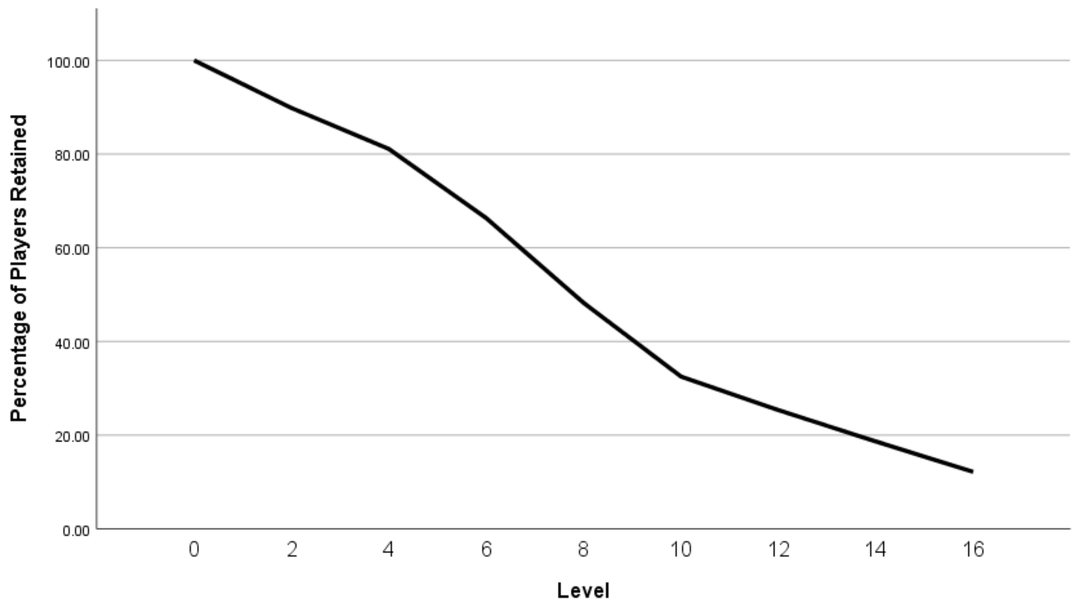


Figure 1.1: Player retention in the *Foldit* tutorial. Data aggregated across 1,957 new players of *Foldit* from the dataset collected in [337]. When these data were collected, the *Foldit* tutorial contained 38 levels, yet fewer than half of the players reached level 8, mere minutes into the game. By level 16 (perhaps one hour into the game), the game retains fewer than 13% of new players. Although there are scarce generalizable statistics on retention for similar games, most mobile games average 25–30% Day 1 retention [297], which is more than twice *Foldit*'s retention, assuming the tutorial is a Day 1 activity (which internal metrics confirm). And *Foldit* is considered one of the most successful citizen science games.

vates citizen science players (see Section 2.1.2). Citizen science games are already good at motivating players to start playing and encouraging the formation of tight communities, such as in the competitions of *Foldit* and the collaborations of *Eyewire* [392, 53, 195]. The toughest barrier, then, seems to be the teaching barrier, or "skill barrier" — the challenge of learning to engage and gain competency with a complex citizen science task.

How do we know that something is wrong with the ECCSG onboarding at all? On the surface, we can look at player retention rates [450, 378]. For *Foldit*, one of the most successful ECCSGs, more than 80% of new players don't even attempt a single scientific puzzle, interacting only with the tutorial [572]. See Figure 1.1 for further data on player drop-offs in Foldit. As another example, in the CSG *Phylo*, 90% of registered users complete less than 25 puzzles, and 42% of players fail to complete a single puzzle [254].

But is this lack of retention due to a poor onboarding experience? By definition, yes.

These metrics represent players that were interested enough in the game's premise that they tried to play, but in some way(s), this interest was lost. Since the two goals of the onboarding are to teach and to motivate, then the game failed at one or both of those goals. Given the significant player drop-off shown in these metrics, I hypothesize that by improving the efficacy of the tutorials in CSGs, we can increase initial retention and subsequent expertise so that players can contribute meaningfully to the scientific purpose of the game (which will be defined differently for each project).

We also know something is wrong with ECCSG onboarding because a recent systematic review of citizen science technologies reported that citizen scientists (users of citizen science projects and players of CSGs) explicitly asked for better tutorials in citizen science applications such as *Zooniverse* and *iNaturalist* [477].

How do we know the problem is not with motivation? Motivation comes into play even before initial participation, since motivation is (by definition) what prompts a player to initially[4] engage with a game. Therefore, if something were wrong with the motivational qualities of the tutorial, we would see CSGs struggle to recruit players in the first place. Yet, this is not so.

One *Foldit* publication alone involved over 57,000 players [99], and CSGs continue to boast active membership. Moreover, there is a wealth of knowledge regarding what motivates citizen science players, and CSGs are well-equipped to meet these motivational goals [241, 114, 229, 556, 513]. On the other hand, there is much less research on gaining expertise in ECCSGs [255]. Therefore, I focus primarily on the teaching goal of onboarding. By helping players overcome the skill barrier, this work increases the accessibility of scientific knowledge and its production in CSGs.

1.1 Problem Statement

This dissertation assumes that the primary failure to retain ECCSG players lies in the lack of teaching efficacy in their tutorials (though other factors affect this). The goal is to identify common themes of improvable areas in design and what challenges in development allowed these problems to exist. In this way, the current work aims to enable future research and development to improve the design of ECCSG onboarding, armed with a better understanding of the challenges to design for. Notably, this must be achieved without negatively impacting existing motivations, otherwise we would still fail to engage and onboard players.

[4]Motivation also encompasses continued engagement with the game, but let us assume that one's initial motivation would persist through the first session, yet we are still seeing significant drop-off mere minutes into the game.

1.2 Approach

Most of my studies sit within human-computer interaction (HCI), and in particular, within games-user research (GUR). In this way, I'm positioning this work as a mixed-methods examination of games as constructionist learning environments [133] and constructivist affinity spaces [180, 460]. I'm investigating the player's experience semantically and latently as part of the psycho-socio-technological network between players, games, and developers [406]. Sometimes this work takes a "Big Q" [262] qualitative approach, often through reflexive thematic analysis [58, 59], but also employs iterative, design-based research and the assessment of game metrics as an interpretation of player behavior and cognition [351].

My trajectory was grounded in several theories of learning, play, and motivation, which informed the methods I chose and the interpretations I reached. In particular, when I proposed this work, I framed ECCSGs as triadic, constructed by three co-dependent perspectives [206, 166, 565, 273, 516]. Triadic game design, as coined by Harteveld [206], conceptualizes games like CSGs as belonging to three "worlds:" Reality, Meaning, and Play.

The Reality is the content, context, or ontology of the project, i.e., the domain. For CSGs, this is their scientific domain, such as protein biochemistry in *Foldit* [97, 99, 100].

Meaning is the semiosis, theory, pedagogy, and training objective — how the domain is framed, what values drive the project, and what the goals are. We could consider the scientific purpose of a CSG to be its Meaning, but for the purpose of onboarding design, the Meaning is onboarding players to understanding the scientific goal and how to play, and motivating them to do so. As I said earlier, I'm primarily interested here in the teaching problem, or the problem of *complex learning* [540]: what is the most efficient (i.e., the cheapest on developers' time and resources and on players' time to learn) and effective way to train an expertise in a complex domain? To this, I'm relying on van Merriënboer's Four-Component Instructional Design (4C/ID) model to ground my understanding of complex learning (see Chapters 7 and 8), because this framework comprehensively captures my pedagogical objectives [540, 539, 542, 536, 537]. As further evidence for this choice, 4C/ID has been praised as the "prototypical model" for instructional game design [224]. Although it does not come into play until Chapter 7, 4C/ID plays a key role in how I ultimately design a new ECCSG onboarding experience.

Finally, Play is the ludic component, the game design, the engagement factor, and the concrete technology that brings the project from an idea to an interactive implementation. Play puts the "game" in citizen science game, but it also puts a focus on the player's experience and interactions. In our case, we want to know how they understand their interactions and what skills they actually need and use to engage with the game. That knowledge might be best captured in the *skill chain model*[5] [95, 134] and connects well theoretically with van Merriënboer's notion of a *development portfolio* [540]. In this way, skill chains allow me to frame the pedagogical intent within the player experience in a way that can translate easily to concrete game design concepts.

Because the goal of onboarding design is to produce a designed artifact (an onboarding experience), this work falls within design science: the iterative and intertwined combination of theory, design, and evaluation [404, 405, 353, 45]. Therefore, the latter studies are structured around theoretically-driven and data-driven designs and their evaluations. Specifically, I use the Tandem Transformational Game Design framework [516, 112] as a guide to implementing triadic design science (the "triple bottom line" [516]). This process is an iterative loop of developing toward all three goals (Reality — improved scientific outcomes; Meaning — improved pedagogy; and Play — improved engagement) simultaneously and synchronously.

Yet, despite using triadic game design as a framing theory here, I intentionally do not integrate this framing into the rest of this work. While triadic design is a useful conceptualization to think about how CSGs are interdisciplinary and multi-purposed, it's not critical to the actual work itself. Rather, it would be more helpful not to assume that CSGs are exactly these three worlds; and in fact, Chapter 4 will provide some contrary evidence against this, suggesting that perhaps it is more practical to think of CSGs in terms of science, software, and design [337]. Therefore, rather than limit my approach by forcing a delimitization, I remain flexible in my conceptualization of CSGs as they interface with players, developers, software, science, design, development, and more.

The primary game I will examine in this dissertation is *Foldit*, for five reasons. First, *Foldit* is the oldest ECCSG, turning 15 years old in 2023. This means that the game and community are well-established and its expertise is known but still difficult to achieve. Second, *Foldit* is, by my account, the largest active ECCSG (by number of daily active users[6]), implying that it would be easier to study *Foldit*'s community than other, smaller communities. Third, *Foldit* is notably complex in its mechanics and difficult to learn [243, 395], making it an excellent candidate for the purpose of the current studies.[7] Fourth, there has been previous and concurrent academic literature

[5] This is explored further in Chapter 7

[6] Estimated based on media reports of registered players in different ECCSGs, e.g., [56, 533, 505], game community activity (e.g., Discord), and personal experiences in various games' chat rooms and forums during the course of my PhD.

[7] See also `https://fold.it/forum/discussion/happy-few-gold-players-in-history/`, in which a player comments on the remarkable fact of another player getting a top score after only 9 months "since, usually in older times, the learning curve for Foldit was about 2 years." This player himself only reached a similar top score after 9 years of play.

studying *Foldit* [114, 395] (even its tutorial [14, 173]), meaning that there is precedent to studying *Foldit* and other work to compare my results to. And lastly, as a practical reason, *Foldit* is the only ECCSG to which I have access to its source code and development pipeline.

1.2.0.1 Reflexivity and Positionality Statement

To this fact, I must be transparent that I and my adviser are developers on *Foldit*. Although my developer role was purely for this research, it remains a bias to note, and especially a bias on the part of my adviser as one of the co-creators of the game. As will be discussed throughout the studies which include *Foldit*, I have taken measures to mitigate this bias, but it must still be mentioned as an unavoidable limitation of this work. Recognizing the sociocultural issues surrounding human-computer interaction research, this work raises concerns of accessibility and inclusion to science culture — who gets to participate in scientific knowledge production and organization? Therefore, I believe it is important to acknowledge the inescapable bias of my and my co-authors' personal positions from which we approached our participants and analyzed their data [295]. My co-authors and I broadly come from affluent Western cultures and are immersed in science and gaming culture, thus giving us a wealth of science capital [19] and biases toward gaming norms. Moreover, I and several of my co-authors are *Foldit* developers ourselves, which influences the way our participants see us; however, when conducting interviews or other participant-centered research, I always approached *Foldit* participants as a researcher rather than a developer and encouraged open, critical feedback.

With this approach and positionality laid out, let me more specifically delineate the research focus.

1.3 Research Areas

There are two research questions which form the basis of my studies. First, **RQ1: how is ECCSG onboarding, especially with respect to the teaching problem, currently designed and perceived?** Secondary questions deriving from this include:

A. What are the existing challenges in onboarding from the players' perspective?

B. What are the existing challenges in onboarding from the perspective of other stakeholder groups: educators, researchers, and developers?

C. How is ECCSG onboarding currently designed?

D. What is unique about ECCSG onboarding that separates it from the onboarding of other citizen science games, other games about gaining expertise (i.e., educational games), and traditional (i.e., commercial) games?

By understanding how ECCSGs are designed and perceived, and what makes them different from other games, we can identify the problems in their onboarding design and infer solutions. But knowing the solutions and being able to effect change are two distinct problems, so the second major question is **RQ2: what design changes are practically effective in improving ECCSG onboarding?** This question is most intuitively tested through design-based research (cf. [579]), so I address this research area by implementing changes in the onboarding of a real, live ECCSG (described later). While there are many theories in which I could ground implementation, I focus on a few regarding complex learning, including 4C/ID's "Ten Steps" [540], skill chains [95, 134], and skill-based cognitive task analysis [467] (described later), because these theories have precedent in related domains. Yet, before I can design a new onboarding for expertise-learning, I first need to understand what that expertise is — this involves eliciting player expertise knowledge from players and developers, understanding how they conceptualize this expertise and the skills that experts use. Like the first research question, I operationalize this research area into four sub-questions:

A. How do players and developers conceptualize player expertise?

B. What design changes are suggested by my theoretical grounding and empirical observations?

C. How effective, as a methodology, is applying a Skill-Based Cognitive Task Analysis (SBCTA) [467] approach to onboarding design in ECCSGs?

D. How effective, as a product, is a re-designed onboarding experience based on the theory and data of this work implemented into a real ECCSG?

Through these sub-areas, I explore which of the problems and solutions identified are practically useful for implementation and what, methodologically, might help us refine the process of improving onboarding design in ECCSGs.

1.4 Significance and Contributions

Taken as a whole, this dissertation produces two major contributions to the academic field of games research. First, I produce a theory of expertise-learning in ECCSGs — how expertise is formed and the barriers which prevent it — which benefits the game-based learning (GBL) community, specifically on the subject of complex learning in games. This theory, brought together in Chapter 9, is the summary of thematic findings from observation and empirical research across Chapters 3–7. Second, I provide methodological insights in Chapters 7 and 8 on synthesizing and applying existing methods of Cognitive Task Analysis (CTA) — namely, 4C/ID, skill chain modeling, and SBCTA — to the novel domain of ECCSG. These insights describe what was feasibly implementable in a realistic setting and I suggest how we might better study and design complex learning in games in the future. This second contribution is of interest to CSG development teams and CTA researchers interested in understanding how CTA can apply to other expert domains. In particular, I argue in Chapter 4 how the creation of ECCSGs is a strongly interdisciplinary task and my insights to the development process in Chapters 4, 7, and 8 highlight a novel area of study for the industrial-organizational (I/O) psychology of interdisciplinary workplaces.

As minor contributions, I produce a definition of ECCSGs in Chapter 2 which constitutes a novel way of thinking about ECCSG design, and in Chapter 6 I provide a comprehensive review of tutorial design patterns across citizen science, educational, and commercial entertainment games, which may be of broader use to game designers and game design researchers. Moreover, the insights to CSG design throughout this work — from Chapters 3–8 — provide valuable lenses for CSG developers to iterate on the design and development of data-centric and expertise-centric CSGs alike. Lastly, this work represents a broad spectrum of methods to approach this interdisciplinary subject holistically. These methods included: A/B feature testing, literature synthesis, qualitative and quantitative self-report collection, qualitative content analysis, reflexive thematic analysis, triangulated ethnography, skill-based cognitive task analysis, close reading, research through design, interviews, focus groups, and various forms of playtesting. In Chapter 9, I briefly reflect on the usage of these methods and how synthesizing them produced insights that would not have been gained from a more narrow approach.

1.5 Thesis Statement

Fundamentally, if there is one takeaway message that I discovered and defend across this dissertation, it is this:

> ECCSGs are a potentially valuable means of solving critical twenty-first-century problems in complex, system-driven, self-contained problem spaces by leveraging novel human expertise, but systemic issues in their design and development are preventing players from getting necessary learning support and stunting the popularity of this form of knowledge production; breaking these barriers and creating expertise relies on four elements: practice, community, simplicity, and explanation.

1.6 Thesis Overview

Here I provide an overview of each chapter, as visualized in Figure 1.2.

In Chapter 1, I've introduced the challenge of identifying problems in onboarding design for ECCSGs. I've described how onboarding has two goals, teaching and motivation, and why I'm focusing on the teaching problem. In Chapter 2, I'll discuss the prior academic literature that relates to ECCSG onboarding design, including CSGs and other related terms, motivation and engagement, game-based learning (GBL), and instructional design.

Then in Chapter 3, I'll start the empirical work by asking "research question 0" (RQ0): is any of this really necessary? If we know what theories support learning and motivation, why can't we just design in a way that uses those theories? This chapter shows an empirical study for why that doesn't work, but leaves us with the insight that a player's background knowledge and interests make a big difference in how they perceive onboarding. This chapter prompts a more thorough understanding of what exactly is happening when a CSG is developed and played, leading us to Chapter 4.

In Chapter 4, I conduct two empirical studies to survey perspectives on the "state-of-play," so to speak, in CSGs. In Section 4.1, I answer RQ1_A by surveying players on their experiences with CSGs, learning that their issues are primarily with scientific communication, instructional design, and software issues. In Section 4.2, I answer RQ1_B by surveying developers, researchers, scientists, and educators on their perspectives. This work produces a description of their needs and challenges and the barriers that exist in production, such as funding issues, ambiguous roles, and tensions

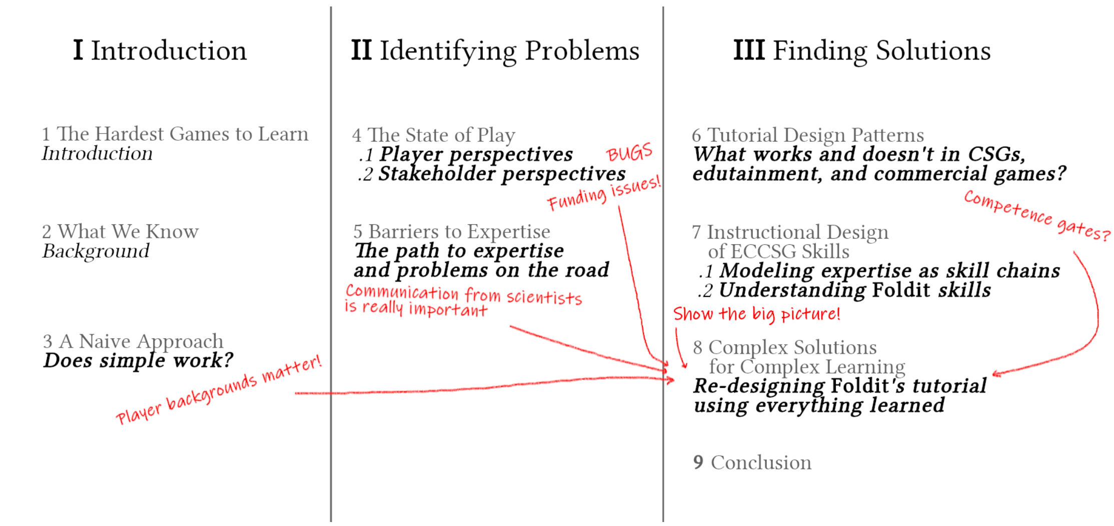


Figure 1.2: Overview of the dissertation by chapter.

between the science and game worlds of citizen science games.

In Chapter 5, I start to address RQ1$_C$ and RQ2$_A$ by determining what the path to expertise is in ECCSGs and what barriers exist along that path. I produce a model of expertise that describes the path as a cycle of exploration followed by social learning, and the major barriers being a lack of instruction, a lack of polish, and a lack of communication.

Chapter 6 addresses RQ1$_D$ — what makes ECCSG onboarding different than in other games? I perform a close reading (or "close play") of over fifty tutorials to make a few generalized statements about what design patterns work or not in various tutorials. Equipped with these design patterns and the insights of Chapters 3–5, I move toward practically implementing solutions to answer RQ2.

I start in Chapter 7 with an analysis of how players and developers understand skill chains as a proxy for how they understand the expertise of ECCSGs. While that study produces some insights into applying the skill chain model, and contributes to addressing RQ1$_C$, it doesn't address the questions of effecting change and the efficacy of doing so. This is answered by my SBCTA approach in Section 7.2, producing answers to RQ2$_B$ and RQ2$_C$. Here, I begin developing a re-design of *Foldit*'s tutorial and describing methodological insights in doing so.

As the final empirical work, in Chapter 8 I summarize the theoretical and empirical insights gathered across previous chapters and fully implement and test a re-design of *Foldit*'s tutorial. This contributes to RQ2$_B$ and answers RQ2$_D$, completing my research agenda. The results, however, do not suggest that my methods were very feasible at all; only a few features were deemed helpful, and empirical data overall showed little noticeable differences in game metrics.

In Chapter 9, however, I reconcile the original theories with the empirical null results and discuss why my approach failed to elicit expertise. Ultimately, short-term GUR methods were insufficient in being able to measure small changes in expertise — a process which often takes months to years. Yet, without these empirical studies, we would not have been able to rule this out, so I stand by my original experimental design decisions.

I conclude with a single sentence which summarizes the theoretical and empirical results gathered across this work: the well-paved road to expertise is a long stretch of practice built by a large community on a few mechanics explained well.

Chapter 2

What We Know

This work stands on the shoulders of centuries of research in psychology and design. Here, I try to organize summaries of the most relevant related work.

2.1 Citizen Science

Citizen science (also called community science [320] or participatory science [202], see Eitzel et al. [148]) is a growing field and model of science wherein volunteers of the general public assist scientific research to produce scientific knowledge [320, 560, 548, 561, 283]. Citizen science is a subset of public participation in scientific research — a superset which also contains crowdsourcing and community-based natural resource management, among others [211, 148, 474]. There are many working definitions of citizen science, which largely describe public volunteer efforts toward scientific work [148]. For the purpose of this dissertation, I am primarily focusing on "science-oriented virtual projects" [559, 70], wherein volunteers participate in goal-oriented scientific activities mediated by information and communications technology.

There are many modes and forms of involvement, ranging from bottom-up —– where citizen scientists themselves initiate the project [214, 258] —– to top-down approaches initiated by professional scientists and research teams. In most cases, volunteers engage with scientific research projects by collecting or analyzing research data [282]. For example, participants can fill in surveys (e.g., the *Dutch flu-tracker* [283]), collect and upload data to a platform (e.g, *iNaturalist* [233]) or play games, such as *Sea Hero Quest* [226]. It is this last group — citizen science *games* — that I focus on.

‡Parts of this chapter were adapted from [332], [338], and an article submitted for review.

2.1.1 Citizen Science Games

Citizen science games (CSGs), sometimes referred to as games with a purpose (GWAPs) [550], arose with the rise of gamification in the late 2000s and were seen as a perspective-shifting technology that redefined online citizen science engagement [358, 131]. CSGs can be either games or gamified projects, and it's worth disambiguating those terms. Although usage varies widely, the most popular definition for gamification in the 2010s comes from Deterding et al. [135] who define it as "the use of game design elements in non-game contexts." What distinguishes a game from a gamified activity includes, among other factors based on one's choice of definition: the psychosocial contexts in which the activity is understood [132], the consequentiality of game actions [247], and the design intent (whether the activity's purpose is gamefulness or whether gameful elements are employed as a strategy toward another goal) [135]. The gamification of citizen science has arisen from a need to motivate a wider audience to engage [401]. To this end, much of citizen science and crowdsourcing has been gamified, and previous literature has reviewed its usage and effectiveness across the field [346, 347].

"Full-fledged" games such as *Foldit* [99] or *Eterna* [287] take an existing problem — protein and RNA folding respectively — and present the problem in a simulated environment in order to enable the creativity of the crowd to solve these difficult problems. Moreover, by turning these tasks into games, the training requirement can become a fun learning curve as novice players attempt to master the game.

Some CSGs instead opt to gamify an existing task. *Eyewire* [514], for example, uses points, badges, and leaderboards (the classic gamification trio [137]) to increase engagement with the cell-coloring task. *Forgotten Island* [403], on the other hand, embeds their gamified moth-labeling task within a narrative-heavy point-and-click adventure game.

Today in 2023, CSGs are being used both for education and research. Several CSGs are used in classrooms or come with separate didactic materials (e.g., on their wiki[1]) or game modes explicitly for education [335]. These games solve real world problems and inspire the next generation of scientists [455, 456, 457, 458]. Thus, CSGs play a crucial role as both a scientific engagement tool and a platform for STEM learning and other online education.

Despite the many ways CSGs are developed and used, we still know very little about what makes *good* CSG design. Although there has been some research on what motivates players to engage with CSGs (e.g., [114, 229, 241]), the player experience and quality of experience in CSGs

[1] `http://eternawiki.org/wiki/index.php5/Educational_resources`

remains critically under-explored [136]. Instead, much of the existing research on CSGs focuses on their overall effectiveness, namely data quality and trustworthiness, and volunteer motivations for engagement [229, 556, 405, 107, 513, 136].

Data quality in CSGs is a concern because games introduce a motivation to play beyond the scientific purpose; this creates a reason for cheating and otherwise transgressive behavior [405, 402]. Furthermore, data quality can suffer from a lack of training and experience [165, 7]. Data quantity is also a concern, but this is mediated by engagement, since more time spent with the project leads to more output.

In short, there are three concerns which affect scientific data quality and quantity: transgressive player behavior, lack of training, and engagement. This dissertation focuses primarily on the lack of training, since engagement is well-researched (see below) and, similarly, transgressive behavior is a known issue in games generally with known solutions[2] (see, for example, [244, 342] and Chapter 4).

2.1.2 Motivation and Engagement

Motivation is a concern because citizen science depends on volunteerism. Additionally, players' motivations for engaging with CSGs are different than one would expect in comparison to entertainment games. Players aren't initially drawn in by the game, but rather by the science — their previous interests in science, the specific research topic, curiosity, and a desire to contribute to research [229, 114, 241, 136].

Continued engagement with the game requires the players to be recognized for their contributions and given feedback, both from the game and from its developers. The players want to feel rewarded for their contributions and know that they are making a difference. Their sustained interest also relies on enjoying the task itself and proper pacing [241]. Furthermore, players are motivated by teamwork in citizen science games, not insomuch by competing against one another (though this has been cited as a motivating factor [425]), but through the establishment of subgroups that each have a clearly defined goal which contributes to the overall project of the citizen science game [229].

This prior research confirms that socialization remains important for engagement. But, rather than achievement being individualistic, CSG players want achievement as a community. Cur-

[2] Here, I don't mean that cheating and similar behavior is completely preventable, but that there are known strategies to mitigating it and that a large portion of the games industry is attentive and active in developing anti-cheat techniques [410].

tis, in her 2015 survey of *Foldit* players, added that citizen science players are also motivated by interaction with others in the community and an intellectual challenge [114]. Players' motivation can also be increased if they share the same goals and values as the project, and enjoy helping others, learning new information, and feeling like part of a team [241, 136]. Other motivational factors researched include narrative [407] and gamification [392, 154, 52], though these both demonstrate mixed results.

Some research has looked at design principles for citizen science projects, and generated several design claims which also apply to engagement with CSGs [515]:

1. **Task Specificity**: Citizen scientists need the ability to discuss aspects of projects and tasks on a platform that scales with the size of the community; lack of integration between discussion and task interfaces wastes effort by hindering communication.

2. **Community Development**: Enabling volunteer moderators with special roles and privileges to filter issues can improve motivation, create leadership, maintain the community, and save time for the science team, but the community still needs timely support from the science team.

3. **Task Design**: Performance feedback and task context improves engagement and motivation.

Overall, there has been relatively little work on the player experience of CSGs, with one notable recent exception [136]. In this study, the researchers identified several exploratory themes of the current player experience of CSGs, including a difficult learning curve, a lack of understanding, and a desire for improved tutorials, all in relation to player engagement. Their results speak to the teaching problem of onboarding which I described in Chapter 1, and demonstrates the need for better training in CSGs and particularly ECCSGs.

2.1.3 Accessibility and Inclusion

Citizen science has proven to be a useful means of scientific knowledge production (e.g., [99, 496, 265]), extending the capabilities of professional scientists with the power of crowdsourcing. However, its mode of production has also raised concerns and critiques regarding, e.g., the exploitation of citizen scientist labor, data ownership, data sharing, conflicts of interest [288, 421, 430, 428], or inclusion and diversity [484, 376].

Who gets to participate in citizen science? The skill barrier that comes with requiring expertise introduces problems for who is allowed to contribute to and benefit from scientific knowledge production [255]. In a case study on the citizen science project *Supernova Hunters* from

Zooniverse [580], the scarcity of data available to classify led to competition; this in turn fostered a bias of participation toward a small cohort of highly dedicated volunteers who were willing and able to compete [484]. The project biased toward males over 65 and biased against individuals who were unable to offer regular weekly time commitments during the work-day. The authors of this case study conclude that cultivating a small, dedicated volunteer community can be associated with a decrease in diversity.

On the one hand, homogeneous groups may outperform heterogeneous groups because in-group bias can create a more positive working experience [75]. On the other hand, heterogeneous groups can have a broader range of skills, knowledge, and opinions; additionally, a lack of diversity can interfere with other project goals such as scientific education and scientific participation [484]. Keep [255] echoes this tension between scientific efficiency and designing for social inclusivity, stating that the skill barrier of ECCSGs forces volunteers to deeply engage to be allowed to participate, despite the fact that most members of the public don't have the resources to commit to such a deep engagement. According to Keep, there are two approaches to democratizing science: the *inclusivity* approach, in which no one is left out of the process of creating scientific knowledge, and the *equality* approach, in which volunteers are deeply involved and treated as equals to the professional scientists.

More concretely, Curtis [115] provides statistics on participation in online citizen science. Reviewing previous literature, she summarizes that participation biases strongly toward males (78% in *Foldit*). Additionally, participants studied were mostly well educated, from developed countries, and were disproportionately involved in IT-related professions. These effects were also seen in *Foldit*; moreover, *Foldit* biased toward older, Western players, with 68% of the players being over the age of 40, and most based in the U.S. and European countries. Curtis also notes that many citizen science projects are biased toward appealing to participants with more *science capital* [19], or cultural and social capital related to science, such as scientific literacy, consumption of science-related media, and more opportunities to engage with science and science culture. She recommends that creating inclusion in citizen science will require addressing the social and cultural barriers, especially with respect to scientific capital.

Given the skill barrier of ECCSGs, there are only two ways to resolve the tension of requiring participants to have expertise: either recruit only volunteers with existing expertise (see the *Argus Project* [168] and the *Polymath Project* [104]) or make expertise accessible to all volunteers in a just and inclusive way. The latter approach may be why data-centric games are designed to have a low skill floor — by reducing the skill barrier, the project becomes more accessible [255]. Ponti et al.

similarly compare *Galaxy Zoo*, which emphasizes equality, to *Foldit*, which emphasizes meritocracy [392]. Their analysis of forum posts suggests that project framing can shape the community's ideals of science and views toward participation and accessibility, notably highlighting the community sense that *Foldit* seems "constrained," "closed and uncertain," and lacking *Galaxy Zoo*'s spirit of belonging and collective contribution.

2.2 Related Terms

Like many academic fields, there is an overlap of jargon terms for various genres and concepts similar to CSGs. Some of these include:

- **crowdsourcing** — solving a specific problem by distributing it to an online crowd [220, 57].
- **educational games** — games with an educational purpose.
- **games for change** — persuasive games with the purpose of creating individual change (in opinions, attitudes, or behaviors) for a specific issue [17].
- **games with a purpose (GWAPs)** — a synonym for HCGs [550].
- **human computation games (HCGs)** — any games which employ human intelligence for real-world problem solving [381].
- **scientific discovery games (SDGs)** — coined by Seth Cooper [97], these games map a scientific problem to a video game which leverages human computation with computer optimization; in other words, an HCG specifically for scientific problems.
- **serious games** — an umbrella term for any game which has a purpose beyond entertainment.

HCGs and SDGs come the closest to what we might think of as ECCSGs, with a few differences. First, ECCSGs do not necessarily leverage computer optimization, as HCGs and SDGs often do (by definition in the case of SDGs) — it would be possible to have an expertise-centric game without the computer performing any optimizations. Second, HCGs don't always require expertise, which ECCSGs do. There are other minor differences given further inspection; for example, in the SDG framework by Cooper [97], scientists present problems to players and aggregate solutions, neither of which are required by the definition of ECCSGs I give below.

One term which I did not include in the list above is Schrier's notion of *knowledge games*, games that "seek to produce knowledge; solve authentic, applicable problems; or generate new ideas and possibilities for real-world change" [454]. Schrier uses this term as an umbrella for citizen science, HCGs, and other crowdsourcing, serious, and educational games, "because they create new real-world knowledge through the playing of the game" [456]. Schrier argues that despite many games being used to solve real-world problems, there is no universal term to describe these games; rather, Schrier found ten different terms used: games for change, engagement games, GWAPs, HCGs, CSGs, crowdsourcing games, social innovation games, knowledge games, social participation games, and games for research [457].

Although I could argue how each of these terms carries different implications for how they solve real-world problems, the point is that CSGs and ECCSGs sit within a field of many different terms for varying approaches to using games for the creation and management of knowledge. The reason Keep [255] and I use the term ECCSGs is to focus specifically on the cultivation of crowd volunteers' expertise (and subsequent production and management of knowledge developed by that expertise) for problems which require complex and creative solutions. This falls strictly within Schrier's knowledge games but only as a subset.

In a more recent publication, Schrier examines Brabham's crowdsourcing typology[3] [57], which seeks to explain the ways crowdsourcing generates knowledge, and provides her own typology for how knowledge games specifically do this [456]. Notably, what I refer to as ECCSGs exist only in the latter two of her four types of knowledge games. First, *algorithm-construction* games, such as *Foldit*, involve complex interactions to support the generation of knowledge or algorithms. Second, *adaptive-predictive* games are a hypothetical design space in which algorithm-construction games feed back into the game system, creating a dynamic loop wherein player data changes the game system and supports a fluid learning environment for continuous data collection and modification. This dissertation seeks, in part, to move toward this hypothetical space by better fitting the game tutorial to the needs of the player, a process which is lately considered dynamic and adaptive [519, 369, 203]. While there are even more terms that could be defined, let us move to the topic at hand: ECCSGs.

2.3 Expertise-Centric Citizen Science Games

ECCSGs were defined by Keep in 2018 as CSGs which task players with gaining a complementary expertise to the domain experts (scientists) and teaching the researchers how to solve

[3]Within Brabham's typology, ECCSGs would likely be classified as "peer-vetted creative production" due to the creative nature of complex learning.

problems that they don't yet know how to solve [255]. In this way, the valuable output is not the problem being solved but the knowledge of how it was solved so that the knowledge can be organized, generalized, and shared [255]. For these games, Keep describes, knowledge sharing and organization are critical processes that require careful translation between the volunteer community and professional scientists.

Notably, to my knowledge all expertise-centric citizen science projects are either games or gamified applications.[4] Researchers choose to frame the (rather complicated) tasks this way because of the motivational power of collaboration, competition, and gamification [241, 114, 229, 556, 513, 425, 51].

But what defines an ECCSG? Although this chapter is mostly on related work, there was no clear definition of ECCSGs prior to this dissertation. Although Keep [255] coined the phase 'expertise-centric citizen science [project/game],' his definition simply requires "volunteers to develop expertise that professional scientists do not have" [255, p. 121], in contrast to data-centric projects which regard providing data to scientists. To further define ECCSGs, I and my co-authors developed a definition [332] in seven criteria which can be used to evaluate whether something can be considered an ECCSG. ECCSGs typically exhibit at least four of the following seven criteria[5] which exist on a qualitative spectrum (for example, an application can vary in its gamefulness). Each criterion will be more thoroughly discussed below.

An ECCSG: (1) is a game or gamified application; (2) requires / develops expertise; (3) makes citizen science contributions; (4) solves novel problems; (5) produces expertise intentionally; (6) produces expertise *concretely* (e.g., by sharing or documenting knowledge); (7) produces expertise *in a new domain of knowledge*. Examples of games and related concepts or genres which meet all or some of these criteria are summarized in Table 2.1. This table is by no means exhaustive or systematically-derived, but captures the most defining ECCSGs of the genre as well as much of the breadth in the gaming and citizen science circles which are closely related to ECCSGs.

Unpacking the ECCSG criteria and related concepts, most notable comparisons to ECCSGs tend to have at least some game elements, including traditional (i.e., data-centric) CSGs,[6] citizen

[4] One reviewer of this statement offered *Stardust@Home* [557] and *Bat Detective* [309] as counter-examples; however, neither of these require expertise. For *Stardust@Home*, researchers explicitly state "even [a] relatively untrained microscopist can readily identify impacts" [557]. For *Bat Detective*, the authors describe their tutorial as a video only one and a half minutes long [309].

[5] Interestingly, at the time of writing no ECCSG strongly meets all seven criteria.

[6] Such as *MalariaSpot* [308], *Project Discovery* [291], *Forgotten Island* [403], *Play to Cure: Genes in Space* [89], and *Phylo* [254]. This category also includes most knowledge games [456], GWAPs [281], HCGs [246], and crowdsourcing games [308], such as *SchoolLife*, *Specimen*, *the Restaurant Game*, *the SUDAN game*, *the ESP game*, *Which English?*, *Who is the Most Famous?* (cf. [456, 454]), and *VerbCorner* [207].

	ECCSGs			Related Concepts			
ECCSG Criteria	**Foldit, Eterna**	**Eyewire, Mozak**	**Decodoku, QM2**	**Traditional CSGs**	**Cit. Psych Sci.**	**Educational Gaming**	**Cit. Sci. Platforms**
Game or gamified	✓	✓	✓	✓	✓	✓	Varies
Requires/develops expertise	✓	✓	✓			✓	Varies
Citizen science contributions	✓	✓	✓	✓	✓		✓
Solves novel problem	✓	✓	✓	Varies	Varies		Varies
Produces expertise intentionally	Partial		Varies			✓	
Produces expertise concretely	✓	Varies		Varies		✓	
Produces expertise in novel domain	✓		✓				

Table 2.1: A comparison of some ECCSGs and related concepts, with their relations to the ECCSG criteria. On the left are games which meet enough criteria to be considered ECCSGs. On the right are concepts and genres closely related to, but distinct from, ECCSGs. This table is meant to be illustrative rather than exhaustive, in order to provide positive and negative examples for each ECCSG criterion. Explanations for each row and column are provided in Section 2.3.

psych-science games,[7] and educational gaming.[8] For comparison, citizen science platforms such as *Zooniverse* [580] and *iNaturalist* [233] represent areas of citizen science which are not games, though some projects on these platforms are ‘gamised’ or gamified [194, 154].

The next two criteria are the most defining for expertise-centric citizen science games, namely the requirements of expertise and of citizen science contributions. Citizen science contributions are simple and objective in definition, and the only example listed that does not meet this criterion is educational gaming. For the criterion of expertise, however, definitions are more nebulous. We operationalize expertise by whether the tasks can be performed competently with very brief training (one could make a specification, such as 30 minutes, but the threshold is moot — data-centric projects are learnable on the order of minutes while expertise-centric projects are learnable on the order of days to years). For example, one of the tasks of *iNaturalist* is to record observations of flora and fauna and guess at their identifications. Although the guesswork may have elements of expertise, most laypersons can take and submit photos for meaningful contributions with only minimal instructions for taking clear shots of the subject and uploading photos. Similarly, most data-centric CSGs and citizen psych-science games are about providing and/or labeling data — through various means — which can be easily taught (e.g., *MalariaSpot* lists its entire instructions on one screen [308]; the *Sea Hero Quest* tutorial takes approximately four minutes [529]). *Eterna*, as a counter-example, lists an 84-page guide [349] among its 46 other player-made guides on the *Eterna* wiki [152]. As of 2021, the *Eterna* tutorial[9] itself contains 118 levels and takes an average player at least 21 hours to complete (personal communication, *Eterna* developer Jonathan Romano, Dec 2, 2021). In this way, expertise is related to *access to contribution*, defined by Rafner et al. [416] as “the likelihood that an average lay person (assuming there are no impediments to participation e.g. physical, socio-cultural, financial, or technological) would make a scientific contribution to the project” — thus, we argue that expertise constitutes a cognitive and temporal barrier to access; the greater the required expertise, the more difficult it is to contribute.

Regarding novel problem solving, this criterion excludes projects which train expertise for the sake of training (novices, AI, etc.) rather than for the sake of new knowledge. For example, educational games are not solving new problems but rather teaching new students about old problems. Similarly, CSGs which focus on natural language processing (such as *VerbCorner*, a gamification

[7] Citizen psych-science is a branch of citizen science wherein the scientific data of study is provided by players about themselves [239], such as in *Skill Lab: Science Detective* and *Sea Hero Quest* [382, 226].

[8] Including both educational games (such as *DragonBox* [476]) and gamified learning (such as *Khan Academy* [345]).

[9] For this dissertation, “tutorials” refer only to developer-made or officially recognized tutorials. *Eterna* is the only CSG I know of which has player-made tutorials, though the concept warrants further investigation.

of labeling verb usage in order to improve the existing VerbNet database [207]) ask players simple questions about natural language. Although the content of the question may be novel, the task is not. In *Foldit* [99], on the other hand, players are designing novel protein sequences through a system of manual manipulations unique to *Foldit* — not even professional biochemistry scientists use this particular workflow [100, 268], and it is possible for an individual player to single-handedly design novel proteins.

Most citizen science and citizen psych-science games, however, fall somewhere in the middle with respect to the novelty of the problem. Games in these categories are often about contributing data to a larger dataset in order to collectively solve a novel problem, such as *Phylo*'s attempts to solve the Multiple Sequence Alignment problem [254]. In this way, these games tackle novel problems by applying machine learning — or sheer amounts of data — to answer a yet unanswered question, though the novelty of the task itself varies.

The final three criteria refine the definition of an ECCSG. To our understanding, there is only one ECCSG to have ever existed which strongly meets the criterion of producing expertise *intentionally*: *Decodoku* [568]. In *Decodoku*, researchers explicitly collected no data except the player strategies via written reports to the scientists [567, 568]. Unfortunately, *Decodoku* had a small audience and is no longer available.

Foldit, *Eterna*, and *Quantum Moves* (QM) [296] (and its sequel, *Quantum Moves 2* [242]) all tie as the second closest to intentionally producing expertise. Their intentions are instead in hybrid intelligence, best phrased by Lieberoth et al. [296, p. 222]: "The aim of Quantum Moves is to combine the best of both worlds in our gamified human quantum optimization: optimization that is rational most of the time, but sometimes makes seemingly random errors or leaps of intuition to rapidly find the sought after solutions." In this way, while these games aim to combine human intuition with computational optimization, there is no explicit intention to harness expertise, for example by codifying player-made strategies with the intention to design new algorithms, although some literature touches briefly on the potential to do this [260, 287].

Next, an ECCSG produces expertise *concretely*, that is to say, in a way which records the expertise by means of sharing, documenting, and organizing the knowledge. Although both *Decodoku* and *Quantum Moves* seek to understand the problem-solving process, there is no public evidence of expertise-formation — they are single-player games, and only the developers have access to the gameplay logs and player reports [296, 567, 568]. Comparatively, *Foldit*, *Eterna*, and

Eyewire are very social games which have extensive player-guides and player-made wikis[10] that continuously update the shared body of knowledge which the players have developed and organized about how to solve the game's problems. Importantly, this is the only ECCSG criterion which distinguishes traditional CSGs, which can be deeply social projects that encourage domain knowledge sharing and organization, from citizen psych-science games, which are inherently about individuals and thus cannot have a similar body of knowledge [239, 114].

Finally, an ECCSG produces *novel* expertise. That is, the expertise that the players develop is a new form of expertise, often complementary to that of the professional scientists who develop the ECCSG but distinct from the scientists' domain. As an example, the *Eterna* player community has developed an entire vocabulary around the idea of "boosting," which is a community-made strategy (unique to *Eterna* players) for stabilizing RNA sequence stems with specific base pair mutations [146]. To employ Gee's Discourse theory [176], the players developed a novel Discourse, complete with its own semiotic domain and highly specific and functional language. Contrast this with learning about RNA in the first place. *Eterna* players must also learn this rather niche knowledge, but details about nucleotide sequencing, the different types of RNA bases, and other declarative expertise is also shared with the scientists who built and run *Eterna*. In this way, *Eterna* can be considered a boundary object [292]: an ill-structured set of work arrangements adapted by two groups cooperating — without a shared definition — as they move between the object's identities and forms.

With these seven criteria laid out, readers may be wondering what types of problems are good candidates for ECCSGs — what are the inherent aspects of a problem which suggest the affordances of ECCSGs? We can determine this by reversing our definition and asking what the affordances are of the ECCSG criteria. Games and gamification afford, among many other properties, self-containment (games are a closed loop with the exception of manuals, wikis, and other paratexts [94]), interactivity (especially system interaction) and motivation to engage in the activity (even to the extent of obsessive passion) [311, 18, 133, 552]. Expertise affords solving complex problems; citizen science affords the collaboration of thousands of laypersons and scientists; and so on. The resulting set of affordances suggests problems which ask a crowd of players to thoroughly explore a system or problem space and become scientists and experts in their own right. The resulting player experience is reminiscent of alternate reality games (ARGs), which often have incredibly complicated problems and require the collaboration of thousands of players across the globe [264, 49]

[10] See `https://foldit.fandom.com/wiki/Foldit_Wiki`, `http://eternawiki.org/wiki/index.php5/Main_Page`, and `https://wiki.eyewire.org/Main_Page`

— the difference being that ECCSGs tackle real subject matter. To phrase succinctly, **ECCSGs are best suited to address system-driven, self-contained, complex problem spaces with many problems to solve**. With these affordances unpacked, we can hypothesize about future ECCSGs that could exist, for example: a game where players succeed by developing increasingly more efficient machine learning algorithms with novel constraints; a game about testing novel agricultural techniques in a simulated environment; or the gamification of translating a dead language or dialect that few people are familiar with.

2.3.1 Scientific Contributions

Citizen science can take a wide variety of forms, and as such, there are many typologies which categorize how volunteer contributions happen. These typologies include, among others: Brabham's crowdsourcing typology [57], Schrier's knowledge games typology [456], Rafner et al.'s framework of citizen science tasks [416], and Wiggins & Crowston's citizen science typology [559]. For the purpose of this dissertation, I define contributions by four components: volunteer action, data origin, project output, and research goal. For example, in *Galaxy Zoo* [164] volunteers classify images of galaxies (action) from telescope imagery (data origin) which creates a database of classified galaxies (project output) for further scientific research on galaxies (research goal). Citizen science projects can have a variety of non-exclusive research goals, ranging from education, to environmental science, to community impact, and more [559]. However, a project's research goals do not impact onboarding design, so I focus only on volunteer action, data origin, and project output. See Table 2.2 for a classification of CSG contribution models.

In this table, data collection from the "environment" includes everything external to the volunteer. Data from the volunteer refers to citizen psych-science [239], described in Section 2.3. For the data origins of expertise-centric projects, "provided" means provided by the researchers (e.g., in *Quantum Moves*, there are specific problems to solve), "selected" means that the volunteers have flexibility in how they choose or solve the provided problems (e.g., in *Foldit* and *Eterna*, players design structures to solve a problem), and "made" means that volunteers create their own problems within the project's domain. Although no projects to our knowledge focus primarily on this type of data origin, one example comes from *Eterna* in which a player used the game to produce their own scientific research [397]. Also note that this typology is not comprehensive and excludes less relevant (to ECCSGs) contribution models such as volunteer computing. For more details on the typologies and classifications of citizen science, crowdsourcing, and public participation in research,

Classification	**Expertise-centric cit. sci.**			**Data-centric cit. sci.**		
Volunteer action	Problem solving			Data analysis	Data collection	
Data origin	Provided	Selected	Made	Provided	Environment	Volunteer
Project output	Dataset or trained algorithm or strategy learning (general or domain-specific)			Dataset or trained algorithm		
Example	*QM2*	*Foldit*	*Eterna*	*Galaxy Zoo*	*iNaturalist*	*Sea Hero Quest*

Table 2.2: A typology of citizen science input/output structures for situating CSG and ECCSG contributions. ECCSGs can uniquely contribute information about how humans solve problems, either generally or specific to the domain of study. See the text in Section 2.3.1 for definitions of data origin.

refer to previous literature that has already thoroughly explored this topic [559, 202, 148, 70].

2.4 Learning in Games

CSGs draw from a history of serious gaming — games developed for purposes beyond entertainment. Most of this history, however, would be pedantic to review here. One key idea of note, though, is game-based learning (GBL). Since the early 2000s, scholars and instructors have tried to apply gamification and gamefulness to learning contexts [399]. The exact definition of GBL varies, but it typically refers to either gameplay with specific learning outcomes or the gamification of learning in general [388]. Many studies in the last two decades have examined learning in games with mixed success [39, 125, 201, 517]. Often, the effectiveness of GBL depends on the execution of implementation [517] and the skills and knowledge of the educators and designers who develop learning games [125].

2.4.1 Expertise

Despite the rich history of learning in games, the development of expertise (the process of learning a skill-set over time[11]) is an especially new area of study [213]. This is true both at the individual level and for peer learning, though there has been some work in games on communities of practice [367], co-production of knowledge (e.g., theorycrafting) [24], habits of practicing action

[11] Contrast this with educational games, which aim to teach a variety of skills, opting for breadth instead of depth.

skills [223], and scientific habits of mind such as "social knowledge construction" [489] (see also the research on "connected learning," or embedding learning within sociocultural networks and motivating it with student interests and purposeful production [235]). The most recent study of individual expertise [213] reported that players gain experience by "*identifying* knowledge and skill gaps; *consuming* and internalising information; *applying* existing knowledge and skills to improve and internalise them...[and *deliberating*] as a meta-process...", with special reliance on paratexts [94] such as gameplay streams and forums.

Only recently has expertise been examined in CSGs [255, 296, 395]. One major observation is that expert players account for a very small percentage of the population but a majority of the contributions and scientific outcomes. Lieberoth et al. [296] refer to expert players as "heroes," mirroring the notion of *whales* in the free-to-play games market, which refers to the small subset of players who generate a majority of revenue in free-to-play games with in-app purchases. Similarly, Stewart, Lubensky, and Huerta identify that crowdsourcing follows a 90-9-1 rule, which they call SCOUT: 90% of participants are (OUT)liers who lurk but don't contribute, 9% are casual (C)ontributors, and 1% are (S)uper Contributors who account for most contributions [490].

Ponti et al. [395] was the first to examine expertise in *Foldit* specifically, and concluded that veteran *Foldit* players develop a "professional vision" (see Goodwin [188]) which guides their decision-making, leading to more manual tuning and usage of external information such as paratexts [94]. In a way, this is similar to the "quiet eye" phenomenon of sports expertise research, wherein expert players have longer gaze fixations on their target, suggesting more fine-tuning during cognitive processing [343]. Finally, Keep [255] solidified the concept of ECCSGs while examining the expertise of *Eterna* players in a longitudinal study. He notes that the skill barrier is especially high for onboarding new players and that a lack of organized knowledge hinders the growth of the project and its community. Moreover, he describes three interconnected phases (neither discrete nor linear) of expertise-centric projects:

1. The professional scientific community translates a problem to the volunteer community, making decisions about problem representation and the problem-solving experience. Challenges include trade-offs between simplicity and accuracy of the problem representation, avoiding misconceptions without creating distracting or unnecessary feedback, and minimizing cognitive load.

2. The volunteer community develops expertise. Challenges include facilitating the development, instantiation, sharing, and organization of knowledge.

3. The volunteer community translates their expertise to the professional scientific community. Challenges include cultural and linguistic translation, the organization of knowledge, and volunteer compensation.

This process conceals a systemic problem in the development of expertise. Because ECCSGs can grow and develop over many years, they create an increasingly complex environment for onboarding new players and raising the skill bar ever-higher for entry to participation. Keep, an *Eterna* developer himself, speaks to this issue:

> *"The project never 'slows' down, so incoming volunteers have to play catch up, while experienced volunteers, who have logged thousands of hours and been doing this for 4+ years are already on to the next idea. This also means there's no way to contribute to the project casually"* [29].

As alluded to earlier, this is part of the teaching problem of onboarding for ECCSGs. So what do we know about solving the teaching problem?

2.4.2 Onboarding Design

As video games are becoming more complex, so too are the techniques which help players learn to play them [509]. Modern tutorial design can be traced back through earlier forms of help including the 'attract mode' of arcade machines, hints, manuals, variable difficulty, checkpoints, spatio-narrative guidance (e.g., mini-maps, control prompts), and finally the interactive tutorials and training missions that are common in today's games [509, 396].

Between the games industry and academic research, much is already known about effective onboarding design for both commercial and citizen science games. This includes principles for 'good' game design (broadly taken as smooth user experiences and intuitive game-user interactions) and GBL. Synthesizing principles from the games industry, human-computer interaction (HCI) research, games-user research (GUR), human psychology research, engagement design, and instructional design, this overview condenses many findings and principles into categorical bundles for brevity and will be unpacked later as needed. To summarize the bulk of these design heuristics, a well-designed (that is, for the purposes of effective onboarding) game should:

- Manage the player's attentional and cognitive resources by reducing irrelevant perceptual and cognitive details and introducing learning materials gradually [398, 40, 113, 259, 500, 294, 558, 215, 328, 325, 156, 563, 248, 471]

- Use both aural and visual channels by presenting verbal information audibly and non-verbal information visually [558, 316, 317, 113, 259, 558, 82, 11]
- Embed learning within meaningful contexts so that instructions are valued and relevant and information provided comes just-in-time (JIT) [459, 458, 235, 177, 558, 537, 494, 497, 549, 259, 549, 40, 471]
- Teach by demonstration, systems exploration, active learning, and cognitive apprenticeship where applicable, challenging and correcting the player's mental model for gradual refinement [91, 92, 558, 40, 423, 497, 259, 562, 259, 144, 540, 537]
- Reinforce learning with mixed and spaced repeated practice at a balanced but varied difficulty [147, 40, 500, 113, 109, 110, 111, 540, 471, 459, 458]
- Personally tailor the learning experience to the player's needs [422, 558, 326, 259, 177, 558, 537]
- Motivate engagement [215, 432, 442, 123, 252, 234, 113, 459, 458]
- Be fun and interactive, not blocking or patronizing [473, 497, 398, 562, 113, 210, 326, 259, 563, 177, 452, 111, 64, 499, 101]
- Provide agency, respect, and comfort [431, 558, 210, 40, 494, 549, 259]
- Teach with real, meaningful whole tasks [537, 539, 540, 494, 326, 549, 113, 40, 563, 459, 458]
- Pass inspections for usability and efficacy, typically done via iterative development and playtesting [215, 40, 563, 398, 497, 326, 237, 549, 259, 563, 119]

Some of these principles have been built into usability and playability heuristics (e.g., [360, 313, 161, 374, 276, 127, 386, 129]). Notably, the GAP (Game Approachability Principles) were developed for designing comfortable new player experiences [130]. More often though, when research focuses on tutorials specifically, studies tend to examine particular variables, such as context sensitivity and whether the game forces the player to interact with new mechanics, rather than examining the holistic experience [419, 14].

Onboarding design intersects with many other theories of learning and motivation in psychology, neuroscience, and UI/UX research and design. For example, Hodent [215] and White [558] both write about factors such as managing the player's cognitive and attentional loads, guiding visual searches, balancing difficulty, scheduling motivating rewards, and personalizing the experience.

Given all of these principles, it is unclear where ECCSG onboarding design is failing these heuristics, or if the known heuristics can explain their faults at all. Therefore, this dissertation attempts to more thoroughly explore how ECCSG onboarding can be improved, using these heuristics as a starting hypothesis.

2.5 Conclusion

Plenty of work has already been done to understand learning in games and what makes CSGs successful. Yet, ECCSGs are still struggling to onboard players. Perhaps ECCSGs are simply not applying the theoretical knowledge already accumulated. In the next chapter, I explore what I call the "naive approach" to solving the onboarding problems of ECCSGs: what if we just apply the theories that we know are successful for effective motivation and learning?
(Spoiler: it doesn't work.)

Chapter 3

A Naive Approach

3.1 Introduction

A game's onboarding serves two goals: to teach the player and to motivate them to play. For CSGs to meaningfully produce scientific knowledge, it is crucial that players are both engaged and competent. As discussed in the last two chapters, there is already a strong theoretical foundation for both teaching and motivating — so why not just apply those theories? Before we dig deeper into the thorny issues of onboarding design, let us try the simplest solution first: directly apply theoretical knowledge.

The first step is to choose which theories to apply. Previous research on improving engagement with games has involved the application of two cognitive design frameworks: **Self-Determination Theory (SDT)** and **Cognitive Load Theory (CLT)**. SDT identifies key factors that are intrinsically motivating to players, while CLT focuses on minimizing mental load, allowing better focus on the game. On their own, CLT is most often applied to formal learning settings, while SDT has seen a range of applications including games. Yet, these two theories combined have not been tested thoroughly in games, and especially not in CSGs. Because CSGs deal with complex topics, such as protein folding [99] and DNA sequence alignment [254], they often require a great amount of mental demand. Moreover, citizen science projects struggle with "drop-outs" [153], suggesting that some players can often feel a lack of intrinsic motivation. However, it remains unknown how well these theories extend beyond their standard usage to apply to the problem of engagement with citizen science games.

I and my co-authors used these theories in a more critical environment than previous

‡Parts of this chapter were adapted from [337]. This work was supported by the National Institutes of Health grant 1UH2CA203780. This material is based upon work supported by the National Science Foundation under grant no. 1629879.

research, applying and operationalizing them to *Foldit*, a citizen science puzzle game about folding proteins (see Appendix A.4). Understanding the game requires a great amount of mental effort, and most of the players' motivations come from a love for science rather than intrinsically from the game itself [229, 114]. We hypothesized that the combination of SDT and CLT to a citizen science game will improve engagement, since CSG require both learning and motivation.

In order to test this hypothesis, we operationalized SDT and CLT to create six experimental features which we tested with new players of *Foldit*. These features are: (1) reducing perceptual clutter (*Minimalist UI*), (2) imposing a minimal-load order on tasks (*Ordered Levels*), (3) providing meaningful choice in task order (*Branching Levels*), (4) encouraging curiosity through detail (*Science Info*), (5) teaching instead of telling (*Strategic Instructions*), and (6) providing context-insensitive help (*Help Panel*).

After implementing and deploying these features to new players of *Foldit*, we collected both game metrics data and self-report survey responses on demographic and psychographic information. Analysis of the game metrics data revealed two findings: first, that *Branching Levels* reduced level re-completions, suggesting that the UI discouraged players from returning to old levels. Second, the survey results identified that player expertise had a significant impact on engagement, as suggested by an increase in level completions and number of sessions played. Regardless of whether this expertise came from prior domain knowledge or gaming experience, both kinds of experience correlated with a significant rise in engagement with the game; conversely, not having this expertise correlated with below-average engagement. This finding agrees with some of the more recent research that personalization is key to properly operationalize theories of learning and gamification [519, 369, 203].

This study offers three major contributions. First, we document the process of operationalizing SDT and CLT as concrete design objectives and then creating and implementing features into a CSG based on these design objectives. This study uniquely applies these theories toward new game features instead of using them to measure an existing game design, and we describe both the generalizable process as well as the difficulties encountered and how this impacts future CSG design. Second, we found that user interfaces, particularly our level selection screen, can affect how players progress through the game. In our case, by guiding players toward the next incomplete level, our new selection screen reduced level re-completions. Lastly, we present strong evidence that expertise, either in gaming or domain knowledge, correlates with game engagement, perhaps because both expertise and engagement are driven by the same underlying motivation.

3.2 Background

How do players engage with a game? First, the player must be intrinsically motivated to engage; then they must be taught the relevant skills needed to play; and finally the gameplay must match their needs and expectations, which covers a range of design considerations from difficulty and rewards to choice and purpose.

3.2.1 Self Determination Theory

Much of our understanding of player motivation has come from SDT and its sub-theories, such as Cognitive Evaluation Theory and Basic Psychological Need Theory [121, 442, 543]. Namely, video games provide intrinsic motivation, rather than extrinsic motivation, by giving the players **competence** in the game mechanics, **autonomy** in the execution of their actions, and (to a lesser extent [122]) providing a sense of **relatedness** to others [441, 123, 409, 149]. The theory suggests that it is this satisfaction of psychological needs which makes video games more intrinsically engaging than other activities. In adapting SDT to video games, researchers developed the Player Experience of Need Satisfaction (PENS) model [432], which measures basic need satisfaction in games based on SDT. (See [527] for a full review of SDT applied toHCI and games.)

However, before a player can demonstrate competence, they must first learn the game mechanics enough to understand and master them. This can be difficult, since by their nature serious games often involve more mentally challenging tasks than other games. Because of this, the road to competence is barred by the player's limited *cognitive load* capacity—the player can handle only so much information at once. Before competence is possible, the player must learn the game's rules, but this process can be made easier if the game's design minimizes cognitive load. That is, in this conceptual model of player engagement, we use CLT to establish better competence, then use SDT as the primary model of motivation.

3.2.2 Cognitive Load Theory

CLT is a robust framework for describing how learning and processing are occurring. This makes it a useful tool for game design since it describes what mental difficulties the player will experience and how to alleviate those difficulties. Cognitive load is pivotal to player engagement because overloading the player mentally will cause frustration and a general negative affect, which can cause some players to disengage or otherwise stop playing [366, 452].

Learning, according to CLT, is the process of chunking individual elements into a *cognitive schema* in order to reduce the usage of a limited working memory [500]. Qiao et al. [411] define cognitive schemata as highly organized and domain-specific knowledge which reduce element interactivity. This is ultimately what differentiates experts from novices for that subject. A schema is stored in long-term memory and represents the learned concept wholly as one element. Schemata also have faster retrieval speeds than their component elements, require fewer retrieval cues, and can be automatically processed without additional resources or cognitive load [293, 411]. Consequently, the goal of learning is to form cognitive schemata. Barriers to this process are what create the difficulties that we are trying to remove in order to reduce cognitive load and improve learning. Cognitive load, which is perhaps best imagined as the demand on the learner's resources, is itself split into three categories: extraneous, intrinsic, and germane.

Extraneous cognitive load represents the inefficiencies in the method of presenting material [294]. Specifically, if a presented element cannot contribute to the formation of cognitive schemata, its presentation adds extraneous load. For example, if this sentence does not help you understand CLT, then it adds extraneous cognitive load by forcing you to process non-essential information.

Extraneous load often results in decreased performance, as shown in an experiment by Muth [350]. In this experiment, students who were given extraneous information on mathematical word problems performed worse than students with either no extraneous information or students who were told before the experiment that extraneous information would be present. It follows, then, that most uses of CLT have historically emphasized the reduction of extraneous load, rather than trying to manipulate the other two categories [500]. Extraneous load can take many forms, ranging from jargon to poor arrangement of material (e.g. from complex to simple rather than reversed). However, in some cases the "jargon" is central to the material, and other times not. (Imagine, for example, the word 'importunate' used in the context of a newspaper article, where another word would suffice, as opposed to the context of a vocabulary lesson, where the word itself is the critical information.) In this way, identical elements may be intrinsic or extraneous based on the learner's needs [500].

There are two other parts to cognitive load in CLT: **intrinsic load**, which represents the inherent difficulty of the interactions between elements, and **germane load** (more recently, germane processing [192]), which represents the cost of learning (i.e., chunking the interacting elements into a cognitive schema) [294, 500, 411]. However, most applications of CLT focus on extraneous load since it is easier and more worthwhile to minimize and because intrinsic and germane loads are linked to the learning process itself [500]. We believe that this precedent applies to the current study as well because intrinsic and germane loads can be affected only by inherently changing the nature of the material to be learned (in this case, the game mechanics and biochemistry), which we do not want to manipulate.

3.2.3 Usage of SDT and CLT

Although SDT and CLT have been thoroughly studied as frameworks (e.g., [500, 432]), most applications of these frameworks to games either don't make changes directly informed by the theories or don't empirically validate the approach. For example, many gamification studies use SDT to inform their general principles [466] such as by justifying the existing design elements of points, badges, and leaderboards [352]. Some studies, such as Mekler et al. [324] and Sailer et al. [444], empirically test these design elements for efficacy based on the theory, though results have been mixed. Entertainment games sometimes apply SDT as a lens to guide design, such as in *Shadow of Mordor*'s Nemesis system [190]. However, SDT has not been thoroughly studied in directly informing domain-specific mechanics, with the exception of two studies. In the health game *Spa Play* [149], developers used SDT to influence goal setting (autonomy), social connectivity (relatedness), and feedback (competence) among other gamification strategies. In a study by Peng et al. [384], researchers tested features intended to support autonomy and competence in an exergame and successfully increased need satisfaction. In summary, some preliminary work applied SDT in the forms of (1) a lens to guide designing entertainment games; (2) a justification for generic gamification elements; and (3) a design principle to create domain-specific features in serious games for health. The current research extends these findings by creating domain-specific features in a citizen science game which involves more learning than previous applications of SDT.

With respect to CLT, only a few studies have considered the importance of managing cognitive load in the context of game-based learning [263], and most work in this field measures the impact of game features on extraneous cognitive load and learning outcomes (e.g. [208]).

For both SDT and CLT, previous work with these frameworks uses them either theoretically or as a lens to interpret measurements of learning and engagement, such as the NASA Task Load Index (NASA-TLX) [205], the Intrinsic Motivation Inventory (IMI) [443], or through coding interview transcripts [149].

By considering CLT during the development cycle, games can be designed to minimize extraneous load, thus limiting the learner's frustration and enabling engagement. Minimizing cognitive load in this way synergizes with the research on SDT, since reducing cognitive load inherently makes the game easier to use, which in turn promotes competence, one of the key components of SDT. In fact, these two frameworks have been combined previously to great success (see, for example, these theories applied to an online medical course [191]). Both CLT and SDT together inform a holistic view of game design for improving player engagement. By focusing on reducing the player's mental demand and catering to their psychological needs, serious games can more easily and reliably put players in a flow state that they enjoy and engage with, satisfying the purpose of the game.

Yet, no research has observed the effects of operationalizing both theories in a game as this study does. Rather, the concepts are typically used for measuring the impact of one change instead of tailoring the game experience to follow these guidelines of learning and motivation. Our study uses the unique methodology of re-designing features to minimize extraneous load and maximize engagement, rather than using SDT and CLT as measurements of an existing design. We chose this method of intervention to explore the potential effectiveness of this approach, hypothesizing that designing with these frameworks in mind would produce more theoretically-driven (therefore more reliable) results than only measuring the design's effectiveness after production.

3.2.4 Personalized Game Design

Rather than applying these concepts in broad strokes, recent research (e.g., [377, 452, 395]) has discovered that different players, and different game genres, engage differently with certain game elements.

Notably, players seek different goals at different levels of game expertise. Park et al. [377] mined player data of a popular massively multiplayer online roleplaying game (MMORPG) and discovered that players early in the game focus both on achievement and social interaction. That is, for a player to succeed, it is essential that they progress in the game by learning the game mechanics, demonstrating their understanding, and developing relationships with other players. Next, when a player is more advanced in the game, their focus turns entirely toward achievement. In this stage, social interaction is less important, and players seek to master the game. Finally, when a player has reached the maximum level, this finding is reversed; achievement is no longer relevant, and a player's engagement in the game becomes almost entirely social.

In a more genre-agnostic report, Schoenau-Fog [452] conducted a survey in 2011 to determine what categories were most important to players for their subjective engagement. The highest rated categories included achievement (both progression and completion), socializing, audiovisual sensation or stimulation, and experiencing the story. This report validates previous evidence that achievement and socialization are important for player engagement, and adds that players are motivated also by the surface ludic elements, such as art, music, and narrative. In applying these findings to the current study, we surveyed players on their expertise and found that, even in citizen science games, player expertise affects the type and magnitude of their engagement.

3.3 Methodology

3.3.1 Theory Adaptation

Foldit is a citizen science puzzle game in which the player attempts to discover the best fold of a protein [99]. Because *Foldit* is a complex game, it is greatly improved by its tutorial; in fact, previous work on its tutorial increased play time by as much as 29% and player progress by as much as 75% [14]. However, because the design of the tutorial does not take into account the principles of SDT or CLT, this game is an excellent candidate for testing these frameworks.

This study demonstrates a unique methodology: whereas most research citing SDT and CLT use them to measure outcomes of an unrelated variable, here the frameworks are the variables themselves. We define one possible operationalization of these frameworks and then implement features based on our operationalized definitions of these theories. In this way, we adapt the abstract theories into a concrete representation within *Foldit* and measure the behavioral outcomes of this representation.

This section describes not only how we operationalized SDT and CLT within *Foldit*, but also how to generally adapt SDT and CLT into the design of one's own game, regardless of its genre, purpose, or educational value.

There were six operationalized changes: *Minimalist UI*, *Ordered Levels*, *Branching Levels*, *Science Info*, *Strategic Instructions*, and *Help Panel*. The first two, *Minimalist UI* and *Ordered Levels*, were designed to minimize cognitive load. *Branching Levels* and *Science Info* were designed to increase autonomy and motivation respectively, and *Strategic Instructions* and *Help Panel* were designed to improve competence. We confirmed that these features implemented the intended design changes through semi-structured interviews with two experts: an expert in human-centered design and SDT, and an expert player of *Foldit*. These experts reported after two iterations that the designed features meet our operationalized definitions and impact player experience as intended.

Below, we review how each concept of CLT and SDT were operationally defined. Then we expand on the creation of each feature and how our operational definition could generalize for implementation into other games.

For this study, we operationally define extraneous load as perceptual clutter, irrelevant information, and introducing complex ideas before simpler ones. Autonomy is operationally defined as perceived choice by the player, specifically the choices they can make within the game. Purpose is operationally defined as any game element which emphasizes the player's interests, namely science. Lastly, competence is defined as the player's understanding of the tools available to them. Note that we do not examine relatedness in this study due to scope constraints, however, we believe that including both autonomy and competence is sufficient for representing SDT in this study, especially in light of the evidence of Rigby and Ryan [432], who show that relatedness is less correlated overall with desired player outcomes, such as fun/enjoyment, than competence and autonomy for popular genres such as first-person shooters and strategy games.

3.3.1.1 Minimalist UI: Reducing Extraneous Load in the User Interface

In its default version, the first few tutorial levels of *Foldit* display several user interface (UI) elements that both are not introduced explicitly to the player and also not necessary for completing the level. These elements include a control panel, a mode indicator, and a score history visualization. To reduce the extraneous load on the player, these elements were withheld from the early levels of the game and then introduced when the player is more familiar with the rest of the game (see Fig. 3.1 for example).

The generalized goal of this feature is to reduce cognitive load within the game elements, introduce only one mechanic at a time and explicitly so, thus segmenting the learning needs [318]. Many games understand to add in one element at a time, but the UI overwhelms the user from the beginning. CLT would imply the need to remove most of the UI until the user has demonstrated a need for more control and information.

3.3.1.2 Ordered Levels: Reducing Extraneous Load in the Level Ordering

The *Foldit* tutorial previously succeeded in comprehensively covering many of the game elements, but was not structured in a way that minimizes cognitive load. The implemented approach to satisfy CLT is perhaps best described by Leppink and van den Heuvel [294]: the task should begin as simple as possible, minimizing complexity and interacting elements, while giving a high

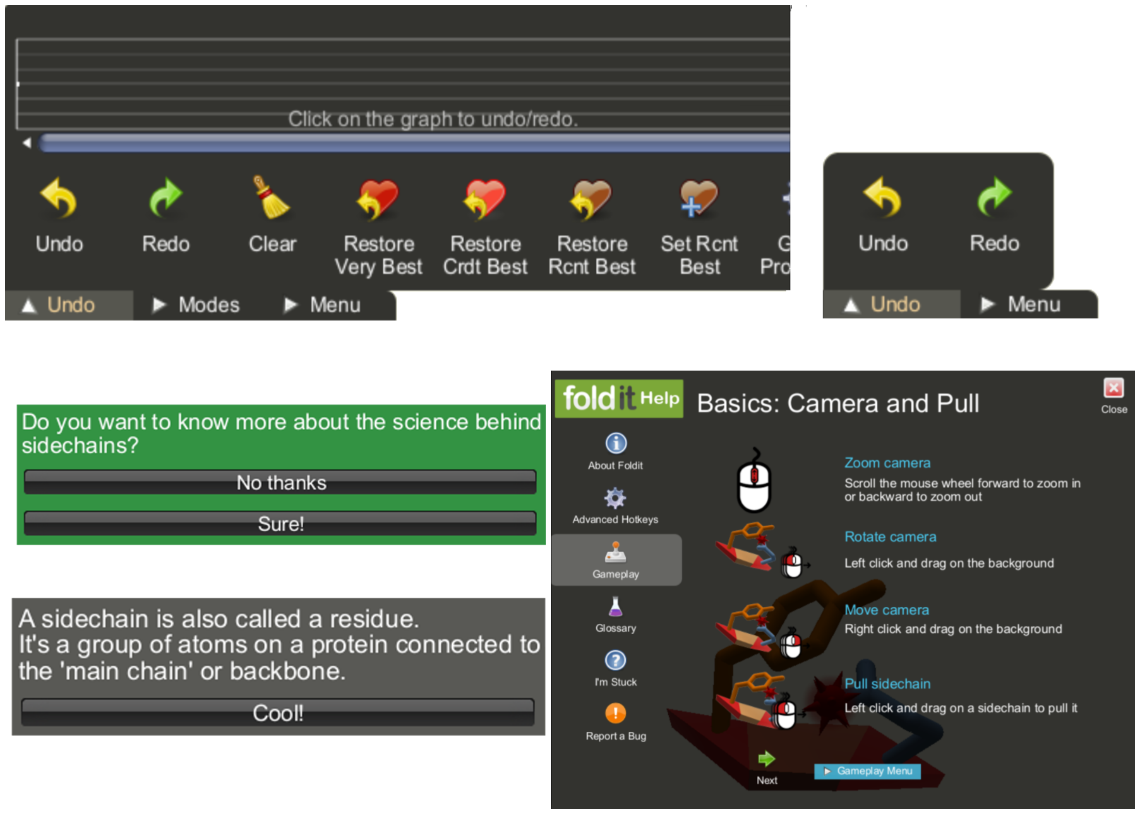


Figure 3.1: Experimental Features. From the top left and continuing clockwise: part of the original UI, that same UI element reduced (*Minimalist UI*), on-demand basic game instructions (*Help Panel*), a pop-up text bubble asking if the player would like to know more about the science of *Foldit* and the following text if the user clicks "Sure!" (*Science Info*).

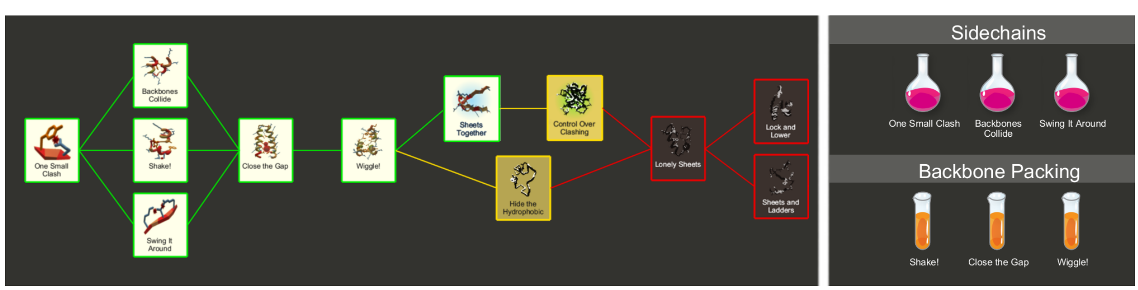


Figure 3.2: Branching Levels. On the right, the original level selection screen is organized into sets and unlocks linearly. On the left, the new level selection screen branches at several points throughout the tutorial.

level of support and instruction. After an element has been introduced, the task should increase in complexity but not interactivity. Particularly, no other elements should be introduced during this process. This method ensures well-ordered problems [177]. Specifically for this study, levels were re-ordered into a linear design based on the researcher's expert understanding of which mechanics should be introduced before others.

To generalize this feature, once the learner has mastered one element, a new one can be introduced with minimal increase in the cognitive load due to the interaction between elements. The concept of element interactivity applies to both actions and elements, and only one should be introduced at a time. During the learning process, support should be gradually reduced in proportion to the learner's proficiency, minding the expertise reversal effect (i.e. redundant support hinders and frustrates experts; see [429]). This process is repeated until the learner reaches a high-interactivity and high-complexity level of play, at which point they can be considered an expert on the subject. For this experiment, levels were simply re-ordered in a way which attempts to meet the criteria described above. In order to use this feature, designers must understand their game's skill dependency tree [95] and use this knowledge to enforce well-ordered problems. Moreover, when a skill is introduced, it should be practiced in isolation against increasingly difficult problems before being mixed with other game mechanics.

3.3.1.3 Branching Levels: Autonomy through Meaningful Choice

Autonomy is afforded by game designs which increase choice, such as choice of goals [443]. To increase the autonomy players felt while playing *Foldit*, those in the *Branching Levels* condition saw a different level selection screen than other players. In this selection screen, rather than choosing levels linearly, the level options branched out, providing several choices for the order in which to complete levels (see Fig. 3.2).

The goal of this feature, more generally, is not only to increase perceived choice, but also to reduce the chance that players will not be able to proceed in the game, since they will have alternate paths of progress available to them.

3.3.1.4 Science Info: Motivation through Purpose and Relevance

When players of citizen science games were surveyed on their experiences, players requested more detail on the science behind the games [229]. By providing more in-game detail to players seeking this knowledge, it is predicted that this will give them more purpose, since scientific contribution is their reported purpose for participating. They will thus become more immersed in the game world, since their newly found purpose within the game appeals to the players' willingness to concentrate [64]. Thus, at several opportunities throughout the tutorial, players in a *Science Info* condition may opt-in to know more relevant information on the science of the game element being introduced in that level (see Fig. 3.1 for example).

Some researchers have argued that Purpose belongs as a fourth dimension to the traditional three-factor model of SDT [160], especially when applied to game-based learning. This internal drive is therefore hypothesized to increase player engagement. The generalization of this feature includes any appeal to the player's personal interests.

3.3.1.5 Strategic Instructions: Context-Sensitive Help

Certain tutorial levels instruct the player on what actions to take, but not why those actions should be taken. For example, in the tutorial level 3-4, "Lock and Lower," the instructions say "Pull this sheet so it lines up with the others. Use [the] Shake and Wiggle [tools] to finish up." However, to increase the player's competence, the instructions should instead describe why sheets should line up with each other, what the Shake tool does, and what the Wiggle tool does. By giving the players more instruction on how to use the tools available to them and less instruction on what actions they should take, the player will become more competent in transferring these skills to new puzzle scenarios, instead of routinely and blindly following directions. From this, we believe that an increased level of competency will lead to an increased sense of competence, though we acknowledge that the two are distinct. In general, for the player to develop a sense of mastery, they must be able to understand the mechanics holistically, rather than as case-by-case usage.

3.3.1.6 Help Panel: Context-Insensitive Help

In 2012, Andersen et al. [14] tested the option of providing a help button in *Foldit* and other games. They found that this yielded a 12% increase in play duration for *Foldit* but caused negative effects in the other games. They concluded that providing assistance gave mixed results which would require further study. However, these results agree with CLT. Because *Foldit* is a more complex game, it demands more cognitive load than the other games tested. Moreover, players working through the early levels of *Foldit* can be considered novices at the game. Thus, these players would benefit from more instruction and guidance [222]. Indeed, in the other simpler games, the assistance is redundant, which causes the players to experience the expertise reversal effect [429]. For this reason, a help button was implemented which provides context-insensitive assistance, such as information on game controls and relevant vocabulary (Fig. 3.1). We predicted that, despite the mixed findings of Andersen et al., a help panel would have an overall positive effect on learning.

More generally, Gee [178] recommends that on-demand help be available for game-based learning, and this is hypothesized to become increasingly useful as the player develops autonomy and mastery over a wide set of precise controls, such as the complex combos in a fighting game or the detailed economies of 4X and city-building games.

3.4 Experimental Evaluation

3.4.1 Experiment Design

New players of *Foldit* (n=1957) were randomly assigned to one of seven conditions, either one of the six experimental features listed above or the control condition. To avoid effects of player interaction, social chat features were disabled for all individuals in the experiment. The data were collected for approximately 6 weeks beginning in November 2018 from the version of *Foldit* available on its website (`https://fold.it/`).

In order to observe the effects of our feature sets, we used the data collection model of Game Metrics Behavior Assessment [351]. In this way, we infer true effects from behavioral differences, such as number of play sessions and level completions.

In addition to game metrics, players in the first level were asked to fill out an optional[1] survey of demographic information, including their level of education, how much prior knowledge they have in biochemistry, and how frequently they play games. Of the participants, 287 users (14.7%) responded to this survey. Note that before playing *Foldit*, all players, whether taking the survey or not, must first consent to a user agreement of data collection based on Institutional Review Board guidelines.

[1] Because *Foldit* is an online game available to anyone and used for more than the purposes of this study (e.g., classroom instruction), we chose an opt-in method for surveying the players. To account for the potential bias in self-selection (e.g., players who answer the survey may have a higher baseline engagement), we restrict analyses that use survey data to only the sample of users who answered the survey.

All Players (n = 1957)			
Factor	Game Metric	χ^2	p
Condition	Level Completions	15.11	.058
Condition	Level Re-completions	46.41	$<$**.001**
Survey Respondents (n = 287)			
Factor	Game Metric	F-value	p
Gameplay Frequency	Level Completions	8.03	**.020**
Prior Knowledge	Sessions	7.45	**.027**

Table 3.1: Summary of main results.

3.4.2 Analysis Methods

First, player expertise was determined along three axes: education, prior knowledge, and gaming, using their responses to the demographic survey (n=287). These were coded as binary variables with high education (n=122) referring to users with “some college” or more, high prior knowledge (n=136) referring to users with “approximately one undergraduate course in biology” or more, and high gaming frequency (n=140) referring to users who play games at least twice per week.

Next, several ANOVA and Kruskal-Wallis omnibus tests were performed to identify the effects of player expertise on our experimental conditions. We considered these game metrics as measures: total time played, number of sessions, unique level completions (i.e., completing a level for the first time), and level re-completions (i.e., completing an old level again). We performed a Kruskal-Wallis rank sum test (n=1957) to identify any main effects of the experimental conditions, and then examined three two-way ANOVAs (n=287) to identify the interactions between experimental condition and a dimension of the player’s expertise (education, prior knowledge, or game frequency). We used an Aligned Rank Transform [566] to adjust for non-normality and performed a Holm correction after the tests to account for multiple comparisons. Finally, pairwise comparisons using the Wilcoxon rank-sum test were performed as post-hoc analyses to understand the nature of significant main effects.

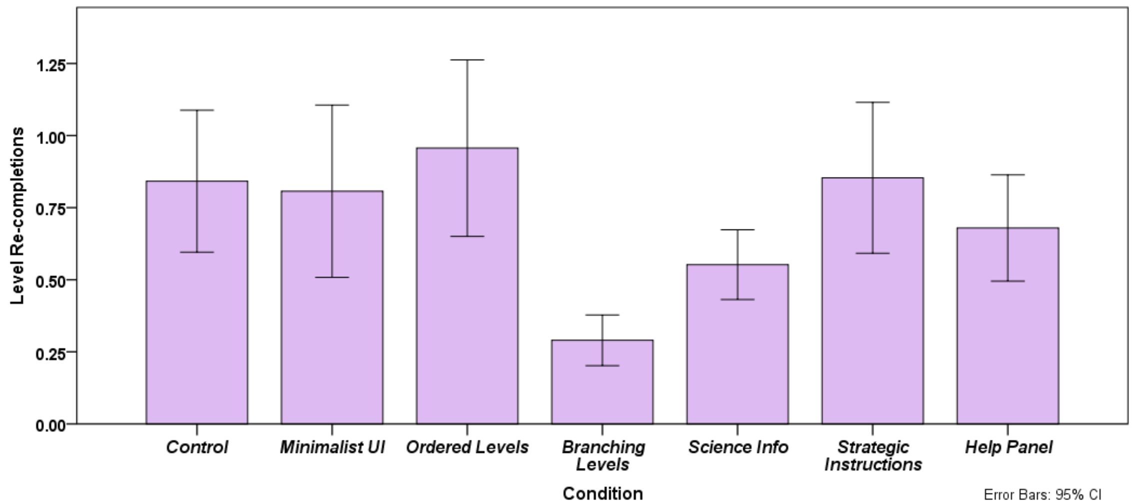


Figure 3.3: Level re-completions by condition. *Branching Levels* significantly reduced level re-completions, possibly because the UI more clearly guided the players toward new levels, such as through the color scheme and by highlighting current and recommended levels. Mean (SD) for each condition: Control 0.84 (2.08); Minimalist UI 0.81 (2.81); Ordered Levels 0.96 (2.47); Branching Levels 0.29 (0.73); Science Info 0.55 (0.97); Strategic Instructions 0.85 (2.22); Help Panel 0.68 (1.59).

3.4.3 Results

For all ART-adjusted ANOVAs (which captured the interactions between player expertise and experimental condition but included only players who took the survey), the main effect of condition was ignored since this information was captured in the Kruskal-Wallis tests (which included all players). The Kruskal-Wallis rank sum test for condition was trending significance after Holm correction for unique level completions (p = .058). A post-hoc Wilcoxon rank-sum test indicated this was driven by the *Strategic Instructions* condition, which had the lowest mean for unique level completions (M=8.84, SD=7.56). The highest mean for unique level completions was *Minimalist UI* (M=10.15, SD=9.65). For level re-completions, the Kruskal-Wallis rank sum test was significant ($p < .001$). A post-hoc Wilcoxon rank-sum test indicated that players in the *Branching Levels* condition had significantly fewer level re-completions (Fig. 3.3); while the control group had an average of 0.84 (SD=2.08) re-completions, players in the *Branching Levels* condition had an average of 0.29 (SD=0.73) re-completions. No significant effects were found for total time played.

Interestingly, two different kinds of player expertise showed a significant increase in engagement (Fig. 3.4). Game playing frequency was found to be significantly correlated with an increase unique level completions (F=8.03, $p < .05$) from a median of 6 levels (low gaming expertise) to a median of 8 levels (high expertise). Prior domain knowledge was significantly correlated with an increase in number of sessions (F=7.45, $p < .05$) from a median of 1 session to 2 sessions. No other significant main effects were found. These findings are summarized in Table 3.1; note that "Level Completions" refers specifically to unique levels completed by the player.

Although no significant effects were found for the *Help Panel*, it was used regularly by players in this condition (n=234). Players clicked it on average 2.92 times (SD=7.45). Similarly, players in the *Science Info* condition (n=215) requested to know more about the science of *Foldit* an average of 3.84 times (SD=4.00).

3.5 Discussion

In this study, we applied SDT and CLT in order to increase engagement and retention in a citizen science game. Unlike previous studies, which used these theories to measure cognitive outcomes, we operationalized these principles in order to construct and implement new game features. This process was described in our methods in a generalizable format which allows other designers to implement similar features for their games. Yet, only a couple of these features had a significant impact on behavioral measures of engagement, i.e., through game metrics data. Although we discuss practical issues in the limitations section below, these data largely suggest that creating effective features by interpreting and operationalizing these theories, at least in an existing game framework, is a difficult process. Not only is it difficult to effect change by operationalizing these theories, it is difficult to measure impact. For example, although we added a help panel to reduce cognitive load, as suggested by CLT, this only indirectly speaks to the theory, and in retrospect, gameplay metrics may not adequately capture the impact of a help panel on cognitive load. We believe that the efficacy (or lack thereof) of these features does not imply that the cognitive and motivational frameworks are wrong. Rather, these principles cannot be easily tacked on as after-thoughts to a game's design. Instead, we expect that these frameworks of learning and motivation can have a significant impact if woven into the design from the start of the design process. Thus, our hypothesis that these features would positively impact engagement was partially supported, as shown by the following examples.

First, the *Branching Levels* condition showed reduced level re-completions (Fig. 3.3). This may be because the UI was more clear that a level was completed, since the *Branching Levels* condition also used color and other visual attention design to indicate completed (green), unlocked (yellow and highlighted), and locked (red and silhouetted) levels. In this way, the UI design en-

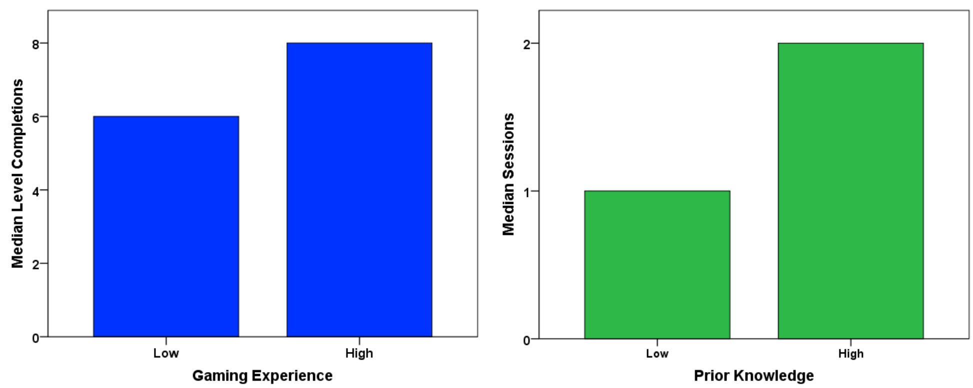


Figure 3.4: Prior domain knowledge and game expertise affect engagement. (Left) Players who play games frequently (median 8) completed on average more levels than players without that expertise (median 6; $p < .05$). Note that the median number of level completions across all players (n=1957) was 7, between low and high gaming expertise. (Right) Players with high prior domain knowledge (median 2) played significantly more sessions than players without prior domain knowledge (median 1; $p < .05$). Both figures show that player expertise correlates with an increase in behavioral engagement. Note that medians are shown to more accurately represent the data which are not normally distributed.

couraged players to progress and visualize their progress, rather than re-complete previous levels. Additionally, since the new level selection screen shows what the level looks like before the player enters the puzzle, this may have reduced confusion and re-completions resulting from players being unsure whether they had solved the puzzle before or not.

The experimental condition with the biggest impact on level completions was *Strategic Instructions*, which was counter-intuitively negative. The change was intended to increase competence by replacing exact commands (e.g. “first Shake, then Wiggle this protein”) with explanations of how to use the tools and why. However, this lack of direct guidance may have led to an increase in cognitive load, since the instructions don’t contribute toward forming a cognitive schema for players who are unfamiliar with *Foldit*, namely all new players in our experiment. Hawlitschek and Joeckel [208] found a similar result when testing whether a digital educational game should include explicit learning instructions. They concluded that the framing added increased extraneous load which led to the observed decreased learning outcomes. In our case, players may have been so unfamiliar with the game mechanics, regardless of background, that more nuanced instructions on their usage were overwhelming.

No significant effects were found in *Minimalist UI*, *Ordered Levels*, *Science Info*, or *Help Panel*, yet these features largely trended toward more engagement through level completions. As future work, we will continue to examine these features, since they may have impacted perceived autonomy and cognitive load, despite showing no significant effects on game metrics data.

Perhaps most notably, players with more expertise engaged more with the game, regardless whether the expertise was a biochemistry background or experience playing games (Fig. 3.4). These axes of expertise increased engagement in distinct ways: frequent game players completed more levels, and players with more domain knowledge played more sessions. This result is unsurprising, since SDT would suggest that the extra expertise gives these players increased competence. Moreover, game players seem to apply their expertise by making more game progress, and players with more domain knowledge seem to have more interest in returning to the game. Critically, this finding emphasizes that *Foldit* is both a citizen science project *and* a game, and invites players from both communities.

This finding also agrees with recent research suggesting that personalization is vital for effective learning and gamification [519, 369, 203]. Since the player experience (i.e., the way the player perceives the game) varies with their expertise, the game's design ought to consider the player's skills and preferences in order to construct the best experience for that particular player. Future work in this field should investigate the efficacy of dynamically tailoring tutorials to the player, not just in difficulty, but in how the material is presented, how rewards are structured, or even how the goal is framed.

3.5.1 Limitations

Although significant effects were found in the applications of these frameworks, the features implemented were relatively small compared to *Foldit*'s larger game design. Since this work was built on the existing game structure of *Foldit* (11 years old during this study), large-scale implementation proved difficult and time consuming. But being mindful of these frameworks at the beginning of a game's design process could address this issue and allow more in-depth implementation of the operationalized design goals. Moreover, the operationalization process itself is a limitation in that there is a layer of abstraction between the theory and the implementation. This adds uncertainty to whether our findings (e.g., in level completions) were directly influenced by the constructs of SDT and CLT, or whether confounds exist in the design. We believe this limitation is acceptable to study theory-driven game development, since the alternative is ignoring the theory in the design process entirely.

The second major limitation to this study is that it examined only one game, and the effects may not generalize to other games or designs. However, this work demonstrates that even small changes to a game's design can have a significant impact on engagement, and this line of research will be worth continuing for the efficiency of serious games in the future.

Additionally, in representing SDT in this study, we do not create any features which capture an operationalized definition of relatedness. However, since autonomy and competence have been shown to have a large impact on similar games [432, 122], we believe that including autonomy and competence is sufficient to approximate the effectiveness of this approach.

3.6 Conclusion

This study operationalized the learning framework of CLT and the motivation framework of SDT to create concrete design objectives. These objectives inspired new features which were implemented into the citizen science game *Foldit* in order to understand the effects of learning and motivation on a complex serious game. Uniquely, this study applied SDT and CLT in order to change the design of a game, rather than using these frameworks to measure the efficacy of existing game features. Moreover, we describe in our methodology how these operationalized definitions can be generalized for feature implementation in other games. In addition to this documentation, we present two other major contributions. First, we found that an updated level selection screen significantly impacted how players progressed through the game. This finding suggests that user interface design guides the player's choices. Secondly, the player's background strongly influences engagement in multiple ways. Since citizen science games are both citizen science projects and games, the player experience is influenced by both previous game experience and prior domain knowledge in the game's subject.

This study is an exploration of theory-driven game design. As opposed to applying traditional game design principles, by operationalizing SDT and CLT for the creation of features, we are ensuring that the design satisfies the learning and motivational needs of the players, whereas traditional game design principles do not capture these needs as directly. Notably, these design implementations were performed on an existing game. This is both a limitation to the methodology, since the existing structure of the game necessarily restricts the design, and a strength, since this study captures the reality that many practitioners have to work within the constraints of existing designs. Our findings would undoubtedly have been different had we used a different operationalized definition of the frameworks or different frameworks entirely, such as flow [110], Information Processing Theory [328], or Keller's ARCS model [256]. Because autonomy, competence, and cognitive load are strongly validated in the literature (see Background) and applicable to design goals of citizen science games, we chose these frameworks and operationalizations as the first step in exploring this methodology.

3.6.1 Takeaways

After trying to implement two of the most successful and thoroughly researched theories in games research [500, 432, 324, 444], not a single implemented feature had the intended effect on engagement. Instead, we found that players' backgrounds mattered: the prior knowledge and interests they were already bringing to the game. Naively trying to implement theories was insufficient, and so, let us take a step back and review the problem space.

In Part II, I focus on identifying the problems to onboarding design in ECCSGs more concretely. What is the state of play, so to speak, for CSGs? What barriers exist to players gaining expertise? By answering these questions, we can better assess exactly where and how players are struggling to understand ECCSGs.

Part II

Identifying Problems

Chapter 4

The State of Play

Despite the large body of research explaining the motivations for CSG players, CSGs still suffer from widespread issues of retention [153, 229, 240]. If we know what attracts and retains CSG audiences, why are CSGs still struggling to maintain an audience?

I approached this question from two angles. First, from the player perspective: what are players experiencing in CSGs, and how do their experiences differ from what the literature understands to be a motivating experience? And second, from the perspectives of other stakeholders: what are CSGs like for the developers and scientists behind them, and for educators and students using them in the classroom?

This chapter is a synthesis of two articles [333, 338] which review the state of CSGs from multiple perspectives in an attempt to understand each stakeholder group's needs and the challenges they face. In the Player Study, my co-authors and I surveyed the citizen science gaming community about their play experiences. In the Stakeholders Study, we interviewed project leads, scientists, developers, and educators analogously about their experiences with CSGs.

4.1 Player Study

To determine the current state of player experiences in CSGs, we sent an online questionnaire to CSG players using a combination of in-game advertisements, social media posts, and game website news posts. This online survey produced 185 valid responses (after filtering) from 9 different citizen science games, though we note a particular skew toward *Foldit* due to its popularity and increased advertising for this survey by the *Foldit* developers. Using qualitative content analysis (QCA), we coded survey responses for commonalities [198, 312]. Among other points, we

‡Parts of this chapter were adapted from [333] and [338].

found that: (1) players are seeking more frequent and clearer scientific communication regarding updates on the projects; (2) players are confused about how to play and need better instructions, (3) user interfaces and controls are often unintuitive, (4) data-focused CSGs suffer from poor task quality, causing player frustration, and (5) CSG software suffers from frequent bugs and crashes that should be addressed.

4.1.1 Methods

Methods were approved by the researchers' institutional ethics board and all participants provided informed consent. Data were collected between April 2019 and May 2021. A total of 237 responses were received and then filtered according to the following criteria: age must be 18-98; responses must specify a valid citizen science game that was being responded for, and duplicate responses were removed. After filtering, 185 valid responses remained; a majority of these (140) were from *Foldit*, while 45 were from games other than *Foldit* (*Eterna*: 14; *Stall Catchers*: 14; *Eyewire*: 7; *Skill Lab: Science Detective*: 4; *Phylo*: 3; *Living Links*: 1; *Mozak*: 1; *Questagame*: 1). We expect the skew toward *Foldit* is due to three reasons: (1) *Foldit* has a much larger active player base than other CSGs [332], (2) *Foldit* recently promoted an Educational mode attracting students and educators [335], and (3) *Foldit*'s developers embedded this survey into their tutorial at a point 16 levels into the game (approximately 1-2 hours of gameplay). Participant ages ranged from 18-78 (M=39.5; SD=17.2). The authors' initial familiarity with these games ranged from passing knowledge to deep expertise; researching and playing these games was done on an as-needed basis for analysis. See Appendix A for details on the games studied.

Open-ended responses were coded using a codebook QCA [198, 312]. Based on recommendations from literature [163], one primary coder wrote the codebook based on a preliminary coding with an effort toward mutually-exclusive codes. Thus, codes were created inductively (data-driven, "conventional") rather than deductively (theory-driven, "directed") [150, 221]. We acknowledge the reflexive nature of qualitative coding, and thus our findings should be considered interpretive, not objective [453].

The codebook was then iterated on through a code-revise-recode process with the other two coders. After five iterations, the codebook stabilized and the three coders proceeded to code/recode the remaining responses. All three coders were authors on the original paper [333]. An intercoder reliability was calculated across all open-ended responses (each question-part treated as a cell and codes measured as present / absent per coder) using Krippendorff's alpha [277], resulting in an alpha of 0.734, which is considered acceptable.

For brevity, I discuss only the high-level results here. See Appendix B for details on the questionnaire and a more thorough report on the results. In short, we surveyed participants on (1) their background, (2) what genres of updates they would like for the game (via ranked choice), (3) their experiences with the game's tutorial, (4) the game's difficulty, and (5) open-ended feedback, which included (a) their favorite and least favorite aspects of the game, (b) requested updates to the game, and (c) their favorite and least favorite aspects of the tutorial (where applicable).

4.1.2 Participant backgrounds

The most salient findings regarding our participants were that they are novices to the game and its topic, play games frequently, and enjoy puzzle and strategy games alongside their citizen science gaming. These results suggest that citizen science games benefit from having well-designed tutorials, reasons to log in daily, and puzzle and strategy elements. Good tutorials are a goal of every game, and most citizen science games already have puzzle or strategy elements. However, little has been done to explore daily login incentives, such as daily quests or bonuses [290]; this may be an interesting avenue to explore for further development.

4.1.3 Update preferences

As described in Figure 4.1, the modal first request from players was more news updates from scientists. This agrees with prior literature that the motivation of contributing to science is one of the most, if not the most, important motivator [114, 120, 145, 153, 154, 229]. Along with scientific updates, new content — such as more puzzles or datasets — was ranked highly among the most participants. This finding suggests that, like many long-standing commercial games, the CSGs we studied follow the “games as a service” model, which relies on continuous content updates to maintain engagement and participation [83, 124].

Bug fixes, quality of life improvements, and new ways to play (e.g., new tools, new game modes) spanned a wide range of rankings. However, a closer look at these responses grouped by player sub-populations (experts, new players, dabblers, etc.) would be necessary to better understand which sub-populations are requesting which updates (cf. citizen science profiling, e.g., [21]). Lastly, updates to social features, story updates, and news from developers were least preferred. The first two of these may be an artifact due to the CSGs studied lacking significant story and meaningful social features (besides basic groups and chat functions), or it could speak to a latent trend among

Figure 4.1: Rankings of update preferences. Science news was the most popular first choice while additional content was broadly favored by the most number of participants. Bug fixes, quality of life improvements, and new modes of play are important to some players but not others, evenly spanning a wide range of rankings. Finally, social updates, story/game updates, and developer news were ranked least important. Updates in this figure are sorted by median rank (white dot), then by the number of respondents who listed that update as their first choice, which is the number in parentheses. The thick black line within the plot describes the interquartile range, while the thin black line extends an additional 1.5 interquartile ranges.

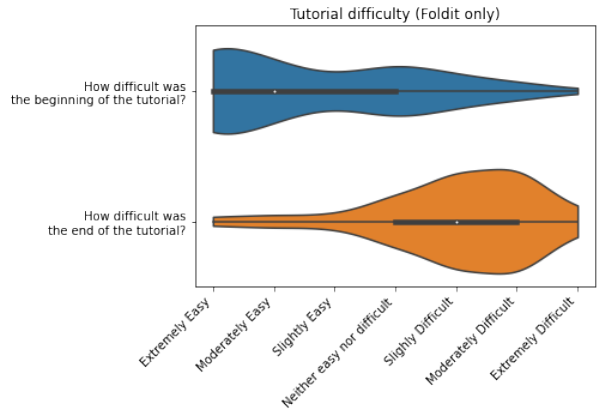


Figure 4.2: Participants found the beginning of the tutorial far easier than the end of the tutorial; a plurality of participants labeled the tutorial as “Extremely easy”, while a similar plurality labeled the end of the tutorial as “Moderately Difficult”. The difference in player responses indicates that work could be done on the various tutorials to smooth the difficulty curve. Few players found any part of the tutorial extremely difficult, though it should be noted that the selection bias in this survey would not capture players who dropped out due to tutorial difficulty.

CSG players that they are more focused on the task and game mechanics than the surrounding community and narrative framing. The fact that players care little for developer updates may speak to the motivation of CSG players to be more interested in the science of the game than the game itself. Alternatively (or in addition), improvements to the software may be seen as less exciting than scientific advances or new gameplay features.

4.1.4 Tutorial experiences

In reporting on the tutorial experiences of CSGs, we are unfortunately limited to describing only *Foldit*'s tutorial. However, we believe this contribution is of value for further consideration of tutorial development in CSGs because several of the themes discussed are agnostic to *Foldit*'s content and mechanics.

As illustrated in Figure 4.2, *Foldit*'s tutorial begins trivially and ends with moderate to extreme difficulty, our participants report. This demonstrates the steep learning curve participants experience in moving from simple controls to the science challenges presented by the game. Participants also note that the tutorial teaches most of the skills needed to play fairly well, though this still leaves room for improvement — and, conversely, room for confusion. Extending the work of Díaz et al. [145], these findings show that both of the CSG tutorials studied in-depth (of *Foldit* in our study and *Quantum Moves* in theirs) had issues with a steep learning curve.

In open-ended responses, participants praised the tutorial for its gradual progression and clear steps, but felt frustrated when the few instructions were insufficient for solving their problem. They suggested that the tutorial could be improved with more examples, more connection to the science topic, and more and better feedback on their performance. Similar to prior work, both CSG tutorials studied have lacked a strong connection to the scientific subject matter which caused players to feel lost or confused at how their play was meaningful [145]. We further found that *Foldit*'s tutorials violated a playability heuristic by taking away the player's hard-won possessions — in this case, the tools they unlocked by completing previous tutorial levels [275]. Other playability heuristics might also be considered violated upon closer inspection, such as having clear goals, balanced challenge, consistent gameplay, and intuitive controls [129, 275].

4.1.5 Game difficulty

With respect to the game's overall difficulty level, we find that the puzzles are mostly engaging though leaning toward moderate difficulty. However, participants were hesitant to look up help, as the plurality of responses indicated that players rarely looked answers up online or asked others for help. This is concerning since there was evidence that some skills were not adequately taught in the tutorial. If players are hesitant to look up help and those skills are not found in the tutorial, then this can lead to those skills never being taught and players consequently feeling stuck.

Our results agree with previous findings of the difficulty of CSGs [145, 255]. Yet, we take this opportunity to ask whether this is where CSGs would ultimately like to be positioned in the space of gaming. This level of difficulty can lead to disengagement or low performance [303, 302]. Moreover, difficulty is a cognitive barrier, much like the logistical barriers of participation that already muddy citizen science participation [255, 484]. These barriers bias participation and dictate who gets to participate in scientific knowledge production and, ultimately, who benefits from it [115, 255, 484].

On the other hand, how much can feasibly be done to make these games easier? The value of some CSGs is employing human cognition and creativity to solve extremely difficult problems; is it the CSG creators' fault for the difficulty of gameplay? We argue yes. Yes, CSG scientists and developers are responsible for lowering barriers to participation of all kinds, especially cognitive ones. As science bears the burden of communicating truth, we must do what we can to make that truth accessible and understandable, enabling participants to engage science and its society-facing problems [523]. In doing so, CSGs must aim to improve their instructional design and scientific communication in order to make even difficult problems accessible to all peoples.

4.1.6 Open-ended game feedback

According to open-ended feedback, one of the primary values of these games is making scientific contributions. This agrees with prior literature on the motivations of CSG players [114, 145, 229, 240]. Moreover, like prior literature, we found that players appreciate the game for having real applications, contributing to scientific knowledge, helping scientists, and feeling like their gameplay matters. Yet, *Foldit* players often described intrinsic game enjoyment more so than making scientific contributions. Intrinsic game enjoyment was coded as the value of the game qua game (i.e., the gamefulness of the experience). Participants enjoyed the games for being relaxing, having aesthetically pleasing color schemes, and simply improving at and enjoying success with a gameful experience. *Foldit* players described, for example, the enjoyment of making a stable protein or an interesting [protein] design, and appreciating the coloring and the game's soundscape. It is perhaps because of *Foldit*'s more pronounced gameful and gamification aspects that intrinsic game enjoyment was the dominant code compared to other games.

Foldit players also commented often on its educational value, which was seen primarily as an "interactive way to see science in action," contrasting static texts and classroom lectures. This is likely due in part to its recent addition of Education mode [335]; however, even before this mode was introduced, *Foldit* has been used by many teachers for its real-time interactivity in teaching biochemistry (e.g., [159]). When this study was published, more than 65 teachers and researchers had contacted or collaborated with the *Foldit* team regarding educational applications (personal communication, *Foldit* team, 2021).

On the other hand, the least favorite aspects of these games were more diverse. Players described confusion, software issues, scientific communication, interface and control issues, and task quality as barriers to their enjoyment, engagement, and productive contribution. For example, participants noted slow feedback on puzzle results and a lack of updates on the research being done based on the game, including publications and progress reports.

These least-favorite results can be seen as a takeaway for what CSGs should focus their efforts on improving. Namely, CSG developers can try to: (1) communicate more clearly and quickly regarding what scientific progress is being made and how players are contributing to it, (2) better teach players how to play, (3) listen to player feedback on interface and controls and collaborate with professional UI/UX designers to effect changes, (4) improve task quality, and (5) fix bugs and crashes (cf. [332]). Although some aspects will look different for each CSG, such as improving task quality, this refinement starts first and foremost with listening to player feedback.

Curiously, the open-ended responses to update preferences did not align with the closed-ended responses. When given the space to elaborate, participants tended to request power user functionality and quality-of-life features. Several times, new players commented that they had no suggestions because they were too unfamiliar with the game to make good recommendations, resulting in expert players dominating the space with their long-lived frustrations and idiosyncratic desires. Thus, "power user functionality / quality-of-life features" was the highest category for *Foldit* and non-*Foldit* games alike, and included for example: features to improve convenience, new interfaces, more access to the internal game functions, new tools, and features which would improve only some advanced workflows.

This finding is similar to the case study of game company Jagex (developers of the MMORPG *RuneScape*) who found that crowdsourcing suggestions from players is limited by which players engage with the crowdsourcing, the shape of ideas they generate, and the aspects of design and development that they value [372]. In our study, not only were most requests limited to features for veteran users, but the remaining requests tended to reflect the participant's least favorite qualities of the game: the UI and controls, the instructions, scientific communication, or bugs and other software issues.

Participants were foremost concerned with the instructional design of the tutorial and secondly with the pacing and structure. For example, participants commented positively that the learning progression was gradual, there were multiple ways to solve the puzzles, and the instructions were easy to follow. On the other hand, the instructions and feedback were sometimes not thorough enough, the tutorial doesn't connect to the real science, and the levels often prevented the use of tools previously given to the player which violates standard playability heuristics [275]. Taken together, these findings suggest that tutorials could be improved by additional just-in-time guidance [178, 471], as well as a more clear link to the science of the game and a better adherence to standard playability heuristics [120, 332].

Across all open-ended participant feedback, the most common codes for *Foldit* were instructions, understanding (or lack thereof) the science of the game, and intrinsic game enjoyment, while for all other games the most common codes were science communication, making scientific contributions, and gamification. The interest of science communication and making scientific contributions is best seen in *Eterna*, as noted earlier regarding *Eterna*'s close connection with scientific feedback and real lab results. When also excluding *Eterna* and *Eyewire* — the two most similar games to *Foldit* — the remaining 24 participants placed gamification as their top concern, followed by software and task quality. These results are notably driven by participants from *Stall Catchers* who requested better gamification, software improvements, and higher video resolution. Together, the overall feedback suggests three core — and equally important — recommendations for improving the CSG player experience: make it about the science, make it understandable, and make it fun.

4.1.6.1 Recommendations

Throughout all participant feedback, their responses highlighted flaws with the current game instruction, both because participants were confused about how to play and because they didn't understand the science of the game, despite wanting to. This agrees with our initial hypothesis that the player experience is one of frustration, and indicates a need for better teaching of the big picture and the science-game loop, or contribution model [334]. This was identified especially in *Foldit*'s tutorial, whose instructions were not thorough enough, not connecting to real science, and violating standard playability heuristics — such as taking away tools the player had earned, inconsistent gameplay, and unintuitive controls [129, 275] — all of which can create further confusion by not meeting standards.

For some games like *Stall Catchers*, gamification was their top concern. CSG teams might consider collaborating with professional game designers to satisfy player interest in gameful or gamified experiences with the task. As reported in Section B.2.1, participants like puzzles and strategy games, so tailoring the task design to those preferences is likely to better attract and retain players.

Overall, these results provide confirmation with previous literature that making scientific contributions remains one of the most, if not the most, important motivating factors for CSG participants [114, 145, 153, 154, 229]. Further, our analysis of participant responses contributes a clearer direction for CSG developers to improve their games, specifically with respect to scientific communication, instructional design, interface and controls, task quality, and software issues. It is important to teach the core gameplay loop and scientific contribution model early (cf. [334]) and iteratively refine your instructions and communication, especially if the project evolves over several years [255]. Scientific communication is critical since it feeds into the satisfaction of making scientific contributions and can also teach and inform players. In this way, communication is the linchpin of CSG success. To this, we suggest quicker, clearer, more frequent, and more regular scientific communication as the single most important aspect CSG developers could focus on. For more details on implementation of these practices, we refer to recent citizen science literature on communication and accessibility [376, 439].

4.1.6.2 Limitations

The most notable limitation of this work is a data skew toward *Foldit* and similar games. However, because our findings are in line with prior work (e.g., [145, 332, 513], we believe that the contributions of these findings remain generalizable to other CSGs. Moreover, our statistical comparisons between *Foldit* and non-*Foldit* responses showed non-significant differences for update preferences and game difficulty (see Appendix B), suggesting that these aspects may be consistent across CSGs.

Secondly, we note that qualitative coding is a trade-off of subjective bias and lack of statistical analysis in exchange for depth and nuance in analysis. Future work would benefit from examining player experiences from a quantitative perspective as well. Yet, this has not been performed to date because embedding the same gameplay data logging technology (telemetry hooks) in all of these games is currently infeasible, and adding the same telemetry hooks in only one or several games runs a greater risk of skew than in the present study.

4.2 Stakeholders Study

Onboarding design in CSGs is not just about the players and the game — the factors that influence how onboarding gets designed and produced exist within a broader context. In fact, the success or failure of the entire CSG depends on a diverse set of stakeholders working together — scientists, players, and game developers. Yet the potential needs of these other stakeholder groups and their possible tensions are poorly understood. Identifying problems in onboarding design is not just about visible issues within the game artifact and the player experience, but includes systemic issues in the larger context of funding, development, and distribution. Although these issues may be more difficult to solve, understanding the context of all agents involved in this network is a requirement for a true understanding of the problems in onboarding design.

In this study, my co-authors and I therefore analyzed the needs and challenges of the individual stakeholder groups of CSGs, which have so far mainly been considered separately or not at all, in order to understand the recurring barriers in existing case studies and active CSGs and provide recommendations on how challenges can be overcome and needs can be satisfied for all stakeholder groups. Our needs and challenges assessment was driven by the following research questions:

RQ1. What are the needs of each stakeholder group involved in CSGs?

RQ2. What challenges does each stakeholder group face? What barriers exist which prevent or hinder their contribution?

RQ3. What overarching factors can be derived from the individual perspectives that most strongly shape and influence the development and maintenance of —– and participation in —– CSGs?

To answer these questions, we conducted a qualitative study spanning two years of ethnographic research and 57 interviews involving stakeholders from ten CSGs: *ARTigo*, *Eterna*, *Eyewire*, *Foldit*, *Forgotten Island*, *Happy Match*, *Reverse the Odds*, *Quantum Moves 2*, *Skill Lab: Science Detective*, and *Stall Catchers*. Using a combination of grounded theory and reflexive thematic analysis, we produce descriptive summaries of the needs and challenges identified for each stakeholder group as well as narrative themes which represent issues involving multiple stakeholder groups. These themes included the ambiguous allocation of developer roles, limited resources and funding dependencies, the need for a citizen science game community, and science–game tensions.

Based on our findings, we synthesized recommendations —– both from our research and directly from our participants —– on how challenges can be overcome and needs can be satisfied for all stakeholder groups. Recommendations include researching previous lessons learned in CSG development (and deciding whether a CSG is right for your project) before creating a new CSG; assigning clear roles to development team members; designing to facilitate knowledge transfer; focusing on community building; and focusing on the entertainment (and gamification) aspects. See Table 4.2 for a summary of recommendations.

CSGs are complex socio-technical "system assemblages" [406] that span many non-human as well as different human actors or stakeholders groups, including, e.g., volunteers/players, scientists, and game designers and developers. Although some studies have examined aspects of the player experience, there does not exist much literature on the scientist's perspective [186, 67] with a couple of exceptions: Golumbic et al. found that scientists' motives and views were often less public-minded than the wider discourse around citizen science. Another study on the "OPAL" (Open Air Laboratories) project in England [430] suggested that scientists were concerned about ethical dimensions on the use of public data and data quality. There are even fewer studies on other stakeholders of citizen science projects, such as policymakers, academics, resource managers, governments, the private sector, or residents of affected environmental areas [358, 434, 319]. Moreover, the development teams themselves are noticeably absent from the CSG literature.

We consider the development team's perspectives critical because they already comprise an interdisciplinary breadth of — and possible tensions between — stakeholders: the project leads, scientists, software developers, game designers, and community managers. Creating a CSG requires expertise in science, game design, software engineering, marketing, communications, and more. Understanding how CSGs operate — from the perspectives of players, researchers, developers, educators, students, and everyone else involved in these projects — can not only contribute to improving models of public participation in scientific research, but also to our understanding of interdisciplinary teams. Moreover, each discipline — each stakeholder group — has unique needs with respect to how they can most effectively contribute to CSG operations.

4.2.1 Methods

This study was an interdisciplinary collaboration, merging a human-computer interaction study conducted by myself and an ethnographic field study by my co-author Libuše Vepřek (see Appendix C.1).

For my methods, purposive sampling was used to recruit researchers, educators, and developers (n=15) involved with a variety of CSGs. Of the participants involved, 6 self-identified as researchers/scientists, 6 as educators, 7 as developers (including game designers and community managers), and 3 as CSG players themselves (self-identifications were not mutually exclusive across roles). Across these participants, the CSGs and citizen science platforms discussed include: *Eterna*, *Eyewire*, *Foldit*, *Forgotten Island*, *Happy Match*, *Reverse the Odds*, *Quantum Moves 2*, and *Skill Lab: Science Detective*. In order to protect the anonymity of participants, no demographic data were collected. Invitations to participate were sent directly via email and linked to a sign-up form for informed consent and interview scheduling. Each interview began with another verbal check of informed consent followed by an hour (on average) of semi-structured interviewing regarding the participant's involvement with CSGs and their experiences, needs, challenges, and advice pertaining to using CSGs. These questions were tailored to the roles they self-identified with; for example, educators were asked "What challenges do your students face when learning how to use [citizen science game]?" while developers were asked "What challenges does your team face in developing [citizen science game]?" Participants were offered a $15 USD Amazon gift card as remuneration. The interviews were audio-recorded and then transcribed for further analysis. In total, 14.2 hours (per participant: M = 56.84 minutes; SD = 31.45) of data were collected. Protocols received institutional review board approval.

To briefly summarize Libuše's methods (detailed in Appendix C.1), she conducted two years of ethnographic research on three CSGs: *ARTigo*, *Foldit*, and *Stall Catchers*. This ethnographic research included participant observation, code, chat and media analysis. She further conducted interviews with developers, project leads, community managers, scientists, and players using purposive sampling.

Prior to the joint analysis, data were analyzed using reflexive thematic analysis [58, 59] with an orientation toward the ways in which each role conceptualizes and frames their relation to the game and CSG community. The analysis approach was primarily deductive, latent, and constructionist. Although no single theory grounded the analysis, it was informed by existing literature on citizen science participation (including, but not limited to, [282, 406, 513, 229, 114, 145, 154, 153, 115, 392]) and Gee's Discourse theory [176] as a framework for conceptualizing Discourse around an artifact (i.e., the CSG).

The preliminary analysis occurred in four rounds of iteratively passing through the data to apply codes, merge codes into themes, and return to the data to validate and refine themes. After four rounds, the preliminary themes included: the vague but valuable CSG niche, unclear tutorials, mismatched expectations, game evolution over years, and several other work-in-progress themes. Yet, when the analyst was producing topics more than themes, he sought to collaborate with other authors to help unpack the meanings within the data and find the key narratives worth detailing. This led to the joint analysis:

In combining our data, Libuše and I discussed the topics and themes we had generated so far and looked for similarities and differences across the preliminary results. This dialogue followed a deductive and constructionist "Big Q" [262] qualitative approach to understand the shared phenomena which descriptively summarize the lived experiences of our stakeholder participants. This included, for example, identifying similarities in described experiences and noting contrasts between how participants describe their intentions versus the media artifacts on their project websites and game systems. Particular attention was given to understanding — theoretically and practically — the lived experiences and interpersonal dynamics of stakeholder groups whose voices are yet unheard in the literature, including third-party researchers collaborating with the first-party scientists involved in CSG development, software developers, game designers, community managers, educators, and students.

We oriented our joint analysis to the research question "What are the needs and challenges of each stakeholder perspective?" As such, our results are divided into two sections: first, we discuss each stakeholder group in isolation, identifying their characteristics and needs. Then, we thematize these findings, highlighting challenges and topics of interest that arose across multiple stakeholder groups and pose issues for CSGs more broadly. Because this dissertation is taking a holistic focus, the individual results can be found in Appendix C.2 while the broader themes and takeaways are discussed here.

4.2.2 Overarching Themes

We found four overarching themes that span across the individual perspectives. These are: (1) roles are ambiguously allocated; (2) limited resources and funding dependencies; (3) need for a CSG community; and (4) science–game tensions. These themes are summarized in Table 4.1.

Theme	**Summary**
Roles are ambiguously allocated	• Developers and participants have vague, overlapping, and/or multiple roles • Overarching visions for the project are blurred by unclear team structure
Limited resources and funding dependencies	• Lack of financial and human resources • Long-term maintenance and basic operations are not supported or funded
Need for a CSG community	• No centralized community for CSGs
Science–Game Tensions	• Issues synchronizing science and game domains
—— Work Environment	• CSGs developed as scientific software • Emphasis on minimum viable product, not enjoyable experience • Game design assumed to be doable untrained

—– Communication	• Communication breakdowns between scientists and game designers due to differing jargon and medium of communication
—– Inherently Difficult Task	• Difficulty integrating science into gameplay • CSG-specific design goals • Requirement of team members to understand multiple domains

Table 4.1: Summary of themes across multiple stakeholder groups.

4.2.2.1 Roles are Ambiguously Allocated

The analysis of the individual stakeholder perspectives has shown that the individual roles within CSG teams are not always clearly distinguishable or distributed and some team members take over several roles at the same time. Moreover, the case studies examined had different team structures of varying sizes and role allocations. For example, some teams included dedicated community management roles, others relied on combined professional scientists and project lead positions.

CSG teams are an interesting example of team structures and collaborative work because they can be located at the intersection of interdisciplinary scientific research teams and science platforms or more specific game development and game platform teams.

As interdisciplinary teams they combine researchers of a specific scientific discipline such as biomedicine and chemistry with computer scientists and game designers who in general work together towards the joint goal of developing and maintaining a CSG. However, the analysis of the individual perspectives has revealed that there sometimes exist different goals within the teams: whereas the developers are focused on building and maintaining a smoothly working platform and game, the researchers's focus lies on scientific knowledge production and the advancement of research.

> *[T]he most [important aspect] from a developer's point of view, I think, [is] first, trying to make the game bug free and make it easy to play and try to make it fun. I think that might be a little bit different from a scientist's point of view. For scientists is probably*

> *the accuracy probably is more important.* [D22][1]

Moreover, the stakeholder analysis has shown that there is sometimes a lack of clarity within the teams about the team structure and the responsibilities that go with individual roles. On the one hand, it remains partly unclear how much of a voice each person gets in executive decisions despite several teams building on so-called "flat" structures within the team instead of hierarchical structures. On the other hand, some team members mentioned their uncertainty regarding who to report to and that they would receive "very mixed messages from multiple directions about [their] jobs" [C9].

These structural ambiguities can make it difficult for individual team members to define their own role within the project. One interviewee described the challenge of "understanding who is responsible for what? Who is capable of what? Where did everybody's role fit in? And then actually define– figuring out my own little space within that" [S8].

Moreover, overarching visions for the CSG in some of the case studies get blurred because of the unclear team structure. There exist conflicting understandings of the vision and current needs. This goes hand in hand with different understandings of the priorities and therefore leads to gaps between the science and design of a CSG, tensions between different roles within the team and communication problems with CSG participants, whose requests to fix bugs in the game are not being prioritized by the developers. Instead of working on the most urgent issue, developers work on what they prefer to work on. [C9] recalls a time when their team brought on a graphic designer who created showy, ostentatious designs which distracted the team from more important bugs and features:

> *[The team would say...] 'This guy made a cool thing. How can we figure out how to incorporate it onto the site?' And it would be like this is literally priority fifty of a thousand. Right now, I don't think we need to be working on this. But that's what would wind up getting worked on or something. [...There would be similar "scope creep"] like, oh, what if we made it do this? What if we made it do that? It's just like literally, please, can we just get this done so that we can go back to other things?* [C9]

In this way, because development is driven by personal developer interest, "a lot of things that happened with [the game] kind of peter off because the leader of that leaves" [ELS15]. Therefore, not only is the scope of the project increasing on unnecessary avenues, work is left half-finished and abandoned.

[1] We refer to statements from our research participants by an identification number and their role(s) as [C]ommunity manager, [D]eveloper, [E]ducation, [G]ame designer, project [L]ead, [P]articipant, and/or [S]cientist. Some quotes are abridged for readability.

Another example described by an interview partner is the different understandings of who to hire: while the developers need immediate assistance, and prefer somebody who is "good enough", the project leads want to hire "the best." Without clear team and project structures, setting the same priorities and pursuing an overarching vision becomes largely difficult and working on the CSG tends to move from issue to issue in an ad hoc way.

The lack of an overarching vision is also reinforced by missing resources (see next section). Since funding is mostly grant-based and therefore driven by projects with a fixed time period, the development of the CSG only moves forward when there is a specific grant to fund a specific sub-project, resulting sometimes in a lack of high-level goals for the project.

4.2.2.2 Limited Resources and Funding Dependencies

No matter how motivated the stakeholders are and how much work they are willing to put into CSGs, the projects always remain strongly dependent on the availability of financial and human resources.

Although, as one of the interviewees expresses in the following, they would continue to work on the projects even without funding, this cannot be sustained in the long run without resources: "I mean there were periods where, where we didn't have funding but that doesn't really matter, we just keep, keep going" [C16].

Hence, the question of resources affects all stakeholders involved, even if not all of them are involved in the acquisition and distribution of resources in their daily working practices. Complaints and concerns about missing resources came up in almost all conversations with CSG team members and can be clustered into a lack of work-hours (developers and their time) and a lack of financial resources.

Most of the analyzed projects faced a lack of team members which would require existing team members to jump from one task to another and to juggle different roles at the same time. Particularly serious for many projects is the lack of developer resources resulting in a backlog of bugs, which get eventually fixed only when exceeding a certain threshold of player complaints. Here it also becomes clear that the participant's requests are not always on top of the priority list of the CSG team:

> *[W]e don't have the manpower to always make everybody happy right now. I wish we did but — and not just manpower but expertise — we only, you know we only have a few people who really know the codebase [. . .] well enough to really develop these things.* [S20]

Although [S20] explains that they "wished" to make everybody happy, the scarcity of resources leads to placing the interests of the CSG team members in the center and the interests of the participants, who devote their leisure time to the project, in the background.

But, as described before, there also exist different goals and priorities within the team which creates challenges when it comes to hiring new developers. The stakeholders responsible for hiring new developers often extend the process to look for the "best," highly qualified candidate, whereas the development team would need and prefer someone good enough who's available. Because of this, [ELS15] describes hiring a developer as "lightning striking" for how rare it is. In reality for the team, however, this process hurts production. Rather than needing a highly qualified developer, the teams just need someone competent who can join quickly and work reliably for a long period of employment [C9] (cf. issues of developer churn in Section C.2.4.1).

Oftentimes, the lack of time and team members or "manpower" [S20] can be translated into the problem of insecure or missing financial resources. Most, if not all, of the projects are mainly based on funding from state agencies, companies, or non-profit organizations. A lot of time must be devoted to writing proposals to obtain grants. One of the biggest challenges, however, is that grants are usually of limited duration and therefore do not ensure long-term sustainability, as one of the project leads explains who had worked on different CSG projects:

> *[S]o there is really not much of [. . .] a very long-term guarantee. [. . .] [T]here is a grant to fund a project, it goes for a few years and then when the grant runs out there is really nothing to support working on a particular project anymore [. . .] [w]hen not some other grants or some other source of funding comes in.* [DL19]

While it is still comparably feasible to obtain funding for new projects, the problem particularly unfolds for long-term maintenance. One of the project leads describes the difficulty of ensuring funding after the initial development phase:

> *[T]here are infrastructure costs, there is still community costs, [. . .] [a]nd we have to do maintenance on the code and [. . .] you know, it costs money to sustain these projects. And I, and I've seen projects disappear like Mark2Cure just because they ran out of funding. So, everybody wants to fund them at first 'cause they are new and innovative but I think the folks who are doing the funding...they just don't think about the fact that, [...] once the project comes to fruition and it's doing what it's supposed to do, it's not like just because it's successful and effective that it makes money just materialize out of the air. You know, it has to be somehow resourced.* [L26]

Most of the CSGs only develop their full potential in the long term by generating research data. This, however, conflicts with the rationale of funding. One of our interviewees has described this problem, which does not only affect CSGs, as “the sustainability problem in citizen science” [L26].

At the same time, funding is often tied to specific scientific projects and established scientific practices such as experiments. As [ES1] states, “developmental resources are placed where people who have money placed them [...] money drives a lot of the developments.” Most often, this does not include funding dedicated to the implementation of CSGs and the improvements of the games: “Saying, ’hey, we want to just improve our platform in general, not tied to any specific scientific outcome.’ That’s hard to get money for” [DP13]. [DP13] explains how the “project lead needs to stretch the scope of a project [in proposing and interpreting grant funding] to cover other necessary features” such as code maintenance. There is often no budget specifically for basic operations, such as code refactoring, bug fixing, porting to other platforms, developing tutorials, or community building. “Like doing just code maintenance is not something you can easily get grant money for” [DP13]. Even something as simple as playtesting for quality assurance is limited on CSG budgets [DGL4, DGL11]. Because of these dependencies, the development of CSGs “is moving at the pace of science funding” [C9].

Due to these difficulties in obtaining scientific grants, some CSG teams additionally turn to other sources of funding as, for example, unrestricted funds of the primary lab or applying for grants that can sneak in maintenance into scientific outcomes. In these practices, development and game maintenance are “wrapped into other things” [DP13]. Another source of funding can be donations, although some of the studied CSGs refuse to take donations from the participants themselves: “it’s like asking somebody who, like, donates blood, hey could you give us five bucks too?” [ELS15].

The lack of resources also includes a lack of public goods [383] available for CSG development. Namely, there are very few existing code libraries publicly available to assist with the common protocols of citizen science gaming. Because of this, developers have to come up with their own solutions and create code from scratch or build on existing CSGs.

> *You don’t really have a lot of people that have worked on similar type of projects so there are not a lot [of] open source tools that you can use [...]. So, most of the stuff I had to code like from scratch. You know like the scoring, the [...] transitions between images, the [...] way the users are giving answers and getting feedback, so all of this needed to be built kind-of from scratch* [D27]

Moreover, developers are often only part-time or inconsistently employed for the project, so these development requirements are done on a volunteer basis — even players will volunteer their time to assist with bug fixes and needed code changes. This volunteerism-by-necessity leads to developer burnout.

> *I stopped partially because — and this is a phenomenon that comes up, I think, frequently in these projects is that — I just got kind of burned out working on it and I was spending a lot of free time on it without..., y'know I was involved in a bunch of different stuff [...] And, spending a lot of time on it and trying to help with [tasks] that I didn't really have any experience doing.* [DS5]

In summary, many of the fundamental aspects of game development — fixing bugs, creating tutorials, playtesting, and building a player community — are, funding-wise, afterthoughts partially solved by developer volunteerism, consequently creating burnout.

4.2.2.3 Need for a CSG Community

Several participants discussed the lack of — and need for — a centralized community specific to citizen science gaming. "At the citizen science conferences," says [DS5], "you don't really feel like you belong." He goes on to describe his challenges trying to explain to others what his CSG is doing and why it's intriguing, since few other citizen science projects share similarities with CSGs. And on the other side, CSGs aren't well-established in the gaming industry either [C9]. They are the hybrid of two worlds and supported by neither.

Yet, there are enough people working in CSGs to form a community. The issue is that the field is fragmented and there is little cross-talk between teams. Participants described that, where collaboration or communication across teams existed, only the project lead was involved and acted as a liaison for the group [ES1, S8]. "I've been part of [project] for six years now," says [ES1], "and in that time I've never been to a conference on citizen science or anything even remotely like it. I don't even know what conferences would be worthwhile going to in this field still." Similarly, [S5], [S8], and [C9] acknowledge that they don't have CSG connections outside of the team. It's "the sort of stuff that the group leader engages in," says [S8].

> *I think that the community being fragmented hurts all of us.* [ES1]

Participants speculated on what value a CSG community would provide. These values included simply having someone else thinking about these issues (such as design issues) to talk to [DS5], discussing pedagogical methods and tutorial design [S8], as well as sharing information, supporting each other, and talking about bigger problems [DP13]. For example, [DP13] suggests it is important to have ethical discussions as a community about the roles of the CSG player. He goes on to consider the idea of stakeholder meetings for the community as a whole (i.e., stakeholders of the field of CSGs).

4.2.2.4 Science–Game Tensions

There are three broad disciplines which work together to create CSGs: science, software engineering, and game design. Software engineering naturally pairs with both science and game design, by the fact that both science and game design fields involve building software. However, science and game design do not mesh as easily, and tensions exist when these two fields “interfere” [139].

This is not the first study to notice this phenomenon, but it may be the first to state it as a team-based dilemma. Winn and Heeter [564] noted a similar tension for serious/educational games, concluding that iterative playtesting allowed development teams to converge on an idea. Within CSGs, Ponti et al. [392] identified tensions between the values of open science and the gameplay of secretive competition, a symptom of this broader dilemma. This tension was confirmed by the study in Chapter 5 [332], attributing this issue to the lack of people who specialize in both the scientific topic and game design, as suggested by Prestopnik and Crowston [400].

But is the problem truly that CSGs need developers specializing in science and gaming? What makes science and game design seemingly incompatible? What is it that creates, as [C9] puts it, a “fundamental push-pull between [...] the science and the game part in ‘citizen science game”’? We further unpack this tension — as a team-based tension — by highlighting three aspects of this dilemma: the work environment, communication, and the inherently difficult task of CSGs.

Work Environment First, the development of CSGs is more aligned with the workflow of scientific research than game development — within development teams, CSGs are framed and developed as scientific software instead of as games. What do we mean by this framing? Game studios operate on tight iterations and frequent releases, while scientific teams are funded by much more long-term grants for very targeted applications.

> *[Scientists are familiar with having] milestones and objectives and stuff like that on paper, but having no practical, real world experience from a program development perspective for, like, how to do this in time for a consumer audience to be happy, because it’s not something that’s familiar in the sciences, because you’re not answerable usually to random humans. You’re answerable to these six months, 12 months, five year plan [...] ’oh, you know, we’ll just plug along at this thing and, you know, we do it when we have the money and we don’t do it when we don’t have the money [...] like, OK. But*

> *actual game studios cannot survive that way... The fact that where I work is essentially a game studio that hasn't released a new title since [approximately ten years ago], and we still exist — in game studio terms, that would be absurd. [...] There's that disconnect between knowing what it takes [...] to sell a game to a user base. And how that doesn't jive very well with laboratory workflows.* [C9]

In essence, CSGs rely on scientific funding mechanisms rather than game-based funding, and this has downstream effects for how development happens and what gets developed (see Sections 4.2.2.1 and C.2.4.1).

The framing of CSGs as scientific software is further confirmed by the project lead's positioning. [DS5] describes that it's not unusual for the project lead to be a scientist and thinking about the project scientifically. [DP12] also notes that their project lead didn't understand the software language they were using for visual display and animation, or that they even understood games. Because all of our case studies start from scientific problems, it makes sense that the project lead would be a professional scientist. Yet, this positioning does not set the project up well to handle the other aspects of CSGs like game design, game development, or public communications.

Finally, the framing of CSGs as scientific software has the downstream effect which we term polish versus possible, described at length by [DS5].

> *There's a difference between creating the software to make it* possible *[their emphasis] to do something and creating the software to make it easy to do something. They [the development team] were good at the first thing, creating it possible, right. But not good at [making it easy].* [DS5]

The scientists on [DS5]'s project were always on the cutting edge. This is the work mindset they understand: solve new and interesting problems, that's what the funding is for anyway.

> *They're trying to figure out: how can we make it possible for people to solve [the latest research question]. That's an important question. But then all of the other stuff they've already done, that's in the past. That's not the cutting edge of research anymore.* [DS5]

Every feature they develop, once made possible, is never polished to be user-friendly (cf. [S20]'s statement that although they can't "make everybody happy," they focus on the happiness of the scientists; for example, players complain about new features being added while major bugs still exist.

> *I'm trying to find a polite word to say, but it's just [...] detritus. You know, like it's just kind of, we've already done that. And that's just out there. And if people want to work through it, they can work through it. But right now, we're focused on [...] the next research angle.* [DS5]

While CSG players don't directly feel the effects of funding sources or project management, they do feel the effects of polish, or lack thereof [332]. CSGs as scientific software are grating, frustrating experiences when players attempt to use them as playthings rather than serious tools, because they were developed with the mindset of creating serious tools, not games. Though we leave this as future work, it is worth exploring in what specific ways the development processes of CSGs differ from projects that are purely scientific or purely games.

The last issue regarding work environments affecting science–game tensions is that CSG teams largely underestimate development of the game aspects. [S8] recounts her experiences with developing for CSGs:

> *What I personally have learned is kind of the complexity of what goes into a good game and the importance of, y'know, I didn't understand the importance of writing good quality code until I started writing games, actually, and I've written a lot of code in my time [...] What I really learned was kind of the value of understanding the story that you're trying to tell and taking into account people's ideas and testing and doing all of that. And I think a lot of this is very obvious to a seasoned game designer. [...] Sitting there going 'Ah, game design, how hard can it be?' And then you sit there and you're drowning in code and you have a game that isn't fun.* [S8]

Even for more experienced game developers, the game aspects of CSGs are still underestimated. [DGL11], for example, emphasized the challenge of designing levels and their pacing. He describes excessive production on his team due to underestimating how much balancing and art would be required for the number of levels they wanted to put into their game. Similarly, [DGL4], in reflecting on his past work, speculates that many aspects of his game could have been better if the team knew more about game design.

Perhaps it is a property of game design that this type of expertise appears easy to replicate. No one would expect to be able to have an intuitive knack for molecular biochemistry or be able to simply "whip together" something in quantum physics. Yet, developers consistently make this assumption of game design — and, to a lesser extent, instructional design (with respect to developing tutorials and educational resources).

Communication The second aspect of science–game tensions is the communication breakdowns that happen within the team. Refer to Figure 4.3 for a diagram of the typical channels of information dissemination to the CSG team and its participants.

According to [S2], "[t]he biggest wall is between the communication of the scientists and the game designers." [ELS15] similarly mentions that communications failed most often with

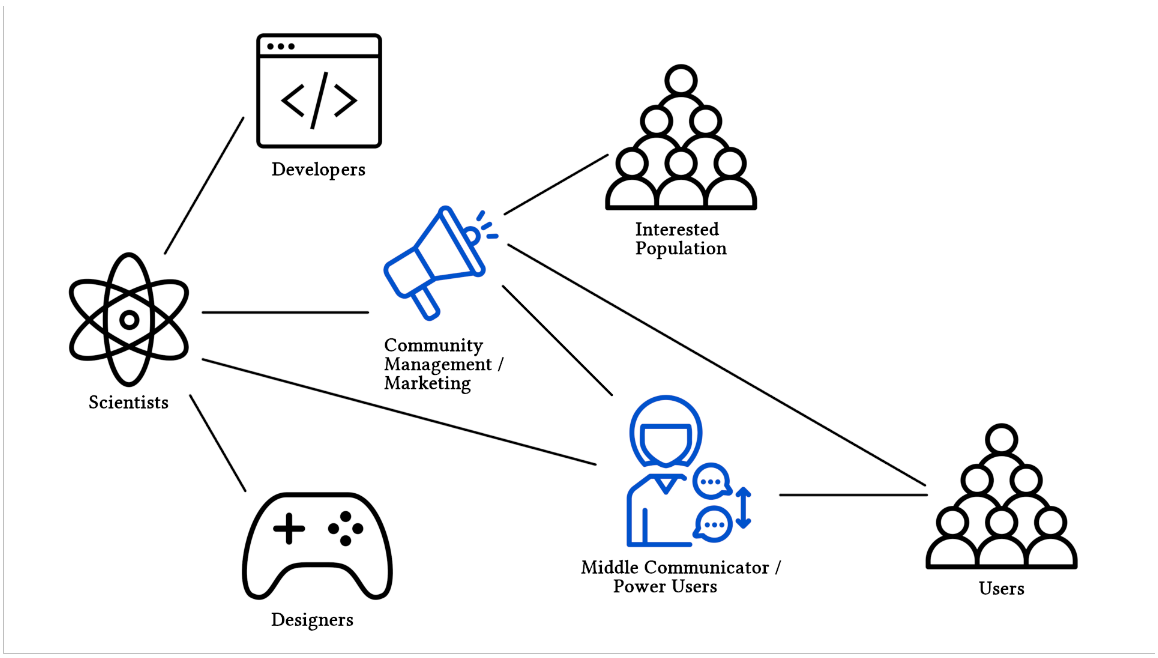


Figure 4.3: Diagram of typical channels of communication and information dissemination. Roles in blue are sometimes absent, in which case limited communication may pass through directly (e.g., from Scientists to Users) or be absent.
Icons from The Noun Project licensed under CC BY 3.0: science by Saideep Karipalli; code by Evon; controller by Abdul Karim; marketing by Tri Sudarti; community by Alzam; translator by Lutfi Gani al Achmad.

design; for example, a scientist would describe a feature to the developer and their meeting would end with a goal in mind, then a few days later the developer would show off their work and the scientist would say “oh yeah, no, that’s not what I meant.”

Similarly, developers struggle to communicate software capabilities and design intentions to the scientists — meaning communication is failing in both directions [S2]. In this way, there is a need for a “middle person” to translate ideas between the scientists and the developers and designers [S2]. [ELS15] similarly describes becoming the “intermediary” with other collaborators, extending the “middle person” role to one which interfaces even outside the team.

Another aspect of communication is whether discussion is online or in-person. [ELS15] believes that having in-person meetings overcomes many differences in jargon: “if you don’t get something or if you misinterpret something, you know, in the next minute, we’re going to figure that out.” Emails, on the other hand, can be misread or misinterpreted, and Zoom meetings are easy to “tune out” [ELS15].

In short, much of the communication within CSG teams is about coming to a shared understanding for its design. This task is made difficult both by the medium — when communication happens online — and by differing sets of jargon, e.g., between scientific terms and game design terms, or even between two scientific backgrounds with differing epistemologies. [S2] gives the example of “model” referring to a cognitive model or a mathematical model depending on one’s background, which can lead to different interpretations of what’s being discussed. Having someone who understands multiple backgrounds and can relate ideas is helpful for mediating discussion in these strongly interdisciplinary teams.

The Inherently Difficult Task Finally, making a CSG is inherently a difficult task because it requires integrating the strongly disparate fields of science and game design. [DGL4] notes that the choice of both the scientific domain and the task are critical for designing a good CSG — some domains and tasks lend themselves well to a game approach, while others are much harder to make into a game. The more specific one’s task requirements are, the harder it is to integrate into gameplay because it becomes a rigid requirement, “You’re basically being handed a [game] mechanic” [DGL4].

Additionally, CSGs come with more design goals than commercial games. CSGs need to design for data collection, data logging, and making clear to players what gameplay counts as scientific contributions. When making the tutorial, developers struggle to teach a domain that’s still being developed — it can be difficult to know what aspects of the game will be important after several years, meaning that the "curriculum" of the tutorial might change [DS5].

The problem of citizen science requiring interdisciplinary expertise has been known for several years. Miller-Rushing describes how citizen science projects require "expertise in science, working with volunteers, education, technology, translation of science to policy — that can be tough to bring together" [339]. But for CSGs, they not only require diverse skill sets but an integration of those skill sets. It's not enough for each team member to take on a skill set, they must share an understanding of the other domains.

[DS5] reflects on how important it was for the software developers to understand the game; it was okay for them to not understand the game if they were focused entirely on code optimization or other software-only tasks, but developing for either the science or the game aspects required understanding other domains. This is especially true for the project leads who need to understand how audiences will respond to a product [C9].

For many scientists and developers getting involved in a CSG, this is their first experience with game development, and their mindset is often still focused on the scientific approach [DGL4]. "This is my first real experience doing [UI design]," says [DS5], "And it was like, [people] do not behave at all like you would expect [them] to." Similarly, [S2] recalls working with the scientists to create game design documentation, mentioning how this act annoyed the scientists because it was too much non-science work for them. Moreover, game design is more challenging than teams expect. Even for teams with game design experience, they still underestimate the amount of effort needed to produce a good game experience [DGL11].

What happens when you have a design task that requires both game design and scientific expertise but don't have experts in both domains collaborating? In short, the result is not fun. "I'm afraid that people are actually starting to relate citizen science games with games with no fun. That is what they already do with educational games," says [S2]. When asked how to fix this problem, they suggested: "A scientist should know that they know about science, but not about design or about development. So getting associated with a group of designers and developers is really important."

Despite science and game design being often in opposition, a good CSG requires them both — combining both science and gameplay "in a way that is more natural and every part of the game feels compelling" [DGL4]. If CSGs are to become better in the future, this will mean changing the work environment and the communication patterns to better support the inherently difficult and inherently integrated design task.

4.2.3 Takeaway Recommendations

In this section, we review 17 Takeaway recommendations for CSG developers (see Table 4.2. These takeaways are given to us by our participants as well as our own recommendations which come from a synthesis of our findings. For other recommendations on CSG development, we refer to previous literature [332, 27, 504].

Takeaway Recommendations for CSG Development Teams
(T1) Learn from previous CSGs — understand the successes and failures before creating a new CSG.
(T2) Consider whether a CSG is right for your project — CSGs are expensive and time-consuming and solve a specific kind of problem; they may not suit your project.
(T3) Come in with resources and a plan — A CSG without secured funding and experts will likely result in failure. Once you have your resources, prototype your technology and build on what other CSGs have done so you don't have to reinvent everything yourself.
(T4) Focus on domains with gameplay opportunities — Some domains are more easily gamified than others.
(T5) Focus on advertising — outreach is critical for building a player community.
(T6) Focus on retention — keep players engaged long-term.
(T7) Focus on community — communicate often, thoroughly, and transparently with users; create a community.
(T8) Have a shared understanding — most team members will need a basic understanding of the science, technology, and game mechanics to collaborate within the team.
(T9) Give clear roles — make sure everyone on the CSG team understands the project and their role within it.
(T10) Defer to the experts — Make sure game designers are clear on the scientific priorities (what has to come before the game design) and have scientists defer to game designers on matters of game design.
(T11) Control various forms of debt — Don't let technical debt accumulate, continuously clean the codebase; continuously improve your developer onboarding, workflows, community management, player interactions, tutorials, etc. These forms of general maintenance will have long-term benefits for scientific research.

(T12) Consider abstraction and gamification first — Take time at the beginning of a project to develop a strong idea of how you will gamify the task and represent it to an unfamiliar audience.	
(T13) Design for socialization and learning — Build features to allow the community to interact and facilitate knowledge transfer.	
(T14) Develop curiosity over education — Aim to build a scientific community rather than a classroom; it is better to have someone engaged and interested in learning more than to dump knowledge on them from the start.	
(T15) Polish really matters — Game polish is critical for engaging players.	
(T16) Publish general takeaways — If you do make a successful CSG, share lessons learned and general takeaways that other projects can benefit from; consider publishing code libraries or open-source resources for other projects to use.	
(T17) Help create the CSG community — Collaborate with other CSG developers.	

Table 4.2: List of Takeaway recommendations for CSG Stakeholders for CSG project success.

First and foremost, we and our participants strongly recommend **(T1) learning from what has been tried in CSGs —– and the lessons learned —– *before* creating a new CSG**. [DP12] says, "There are so many other projects who have actually a lot of experience about gamifying a scientific subject. I would tell them [a new team / new CSG project], go interview those guys, learn from them before you start up. Because many people have been through a number of those things." Similarly, [S8] says, "Understand the literature better [...] not even with regards to reading the papers [...] but understand what different projects have done. Understand sort of where your work fits in with that... There are a bunch of different kinds of citizen science games."

To this point, it is critical to **(T2) reflect on whether building a CSG is the right approach for your project**. CSGs are expensive and time-consuming. Although they can be powerful, they solve a very specific kind of problem, one which is not suited for every project.

> *If you're thinking of doing a citizen science game, you should understand that and then decide if you have the wherewithal to do that. Because there are a lot of ways to motivate engagement in citizen science and games are just one of them. And a lot of the times the citizen science project can be quite successful even if it's not a game project. [...] A game could be a good choice, but it's more work than you might realize. And particularly to do it well, and particularly if you want the game to attract people you wouldn't otherwise attract another way.* [DGL4]

CSGs require a lot of time, money, and expertise. If you are planning on making a CSG on a low budget without the dedicated expertise in science, software, game design, and community management, we recommend you consider other formats for your project. If you do have the resources to make a CSG, first prototype the concept and build on what other CSGs have done before committing to the project in full. **(T3) Approach CSG development with resources and a plan**.

Keep in mind that **(T4) some domains are more easily gamified than others**:

> *If I had a choice about context, I would look really closely at contexts that I thought had more gameplay opportunit[ies] in them. So some of those contexts, I think, like astronomy, I think lends itself well to games. There's a lot of science fiction games that you can do interesting things with astronomy. There is, I think, some kinds of scientific work that lend themselves well to games and other kinds that might be a little more challenging. And I might try to set myself up for success by focusing on a context that seemed like it would work well with games.* [DGL4]

Throughout the development process, there are several aspects to focus on. **(T5) Outreach, such as explicit advertising, is critical for building a player community**. In addition to social media, stakeholders described promotion from influencers (especially non-gaming influencers) and getting on a morning television show as very successful outreach initiatives for their project.

(T6) Focusing on retention and the **(T7) player community** is also critical for long-term engagement and, ultimately, the health of the game. "Player retention is paramount" says [DP12]. Communicating clearly, regularly, and transparently with your users is key for building community. Being transparent with the community — about decisions made behind the scenes and how the science is done — is critical in order to avoid "sound[ing] too promising" [S28] and ensure that participants support, or at least understand, developments of their CSG.

> *Focus on the community. That's what will make [or] break your project. Give them everything you can think of to help them be successful. Listen to their feedback on what they want. Let them surprise you with their out of the box thinking and nuanced realizations and considerations. Communicate often. Make sure that they're able to communicate and collaborate among themselves. Keep a close eye on the pulse and health of the community.* [DP13; post-interview comment]

With respect to the development team, **(T8) ensure that everyone understands the project** and **(T9) their role within it**. Most team members will need a basic understanding of the science, technology, and game mechanics in order to collaborate effectively. [DGL11] emphasizes, "Really understand the science. Really understand the technology and mechanics."

(T10) Use each team member's expertise to your advantage. [C9] says, "Be willing to have the scientists in your projects defer to the game designers about matters of game stuff. And make sure to have the game designers be totally clear on the scientific priorities and what has to actually come before game design."

For development itself, [DP13] says, "Don't let tech debt accumulate." CSGs, which can be decade-long projects with over fifty developers working on them throughout the life of the project, are susceptible to becoming overly complex or hacked-together unless conscious work is put into keeping the code and workflow clean and documented. Additionally, [DS5] adds that "Having players involved in development is really, really good [...] immensely valuable for the research." Player input during design and development can not only help identify user experience problems that the team may have missed, but also provide insights into how players interpret mechanics, mentally model their understanding, and engage with the CSG. These insights are invaluable for bridging the gulfs of execution and evaluation between the player and the game [364].

Ultimately, if you are working on a CSG that you expect to last for years, we recommend **(T11) taking the time and budget now to improve your technology, workflow, developer onboarding, community management, player interactions, and so forth — in other words, managing the accumulating sociotechnical debt**. This polish will improve your scientific output in the long-term [332].

Ongoing maintenance is easier if you start with a strong concept. To this, [DP12] and [DP13] suggest **(T12) putting first and foremost the gamification** (how you gamify the task) **and the abstraction** (how you represent the task to an unfamiliar audience). "Players should be able to come in without any knowledge" [DP13].

Supplement the abstraction with features that let the community teach each other. Since CSGs are, typically, social projects by nature, **(T13)** "**design ways that your community can interact**" **and** "**facilitate knowledge transfer**" [DP13]. This will improve the flow of knowledge from expert players to novices, but also from players to developers. As [ELS15] says, "It's all about communication."

What you're trying to do, according to [DS5], is create a scientific community — not teach students in a classroom. This difference affects how you approach the curriculum design, attempting to **(T14) foster open scientific engagement rather than overwhelm them with technical knowledge**.

In this way, **(T15) the polished experience matters** —– "entertainment really matters" [DGL4]. This was also found by the study in Chapter 5 [332], where I identified game polish as a significant factor for CSGs. Yet, in contrast, we found this perspective highlighted only by developers whose role included game design; other team members were more focused on the scientific goals of the project and less concerned with the player experience. This recalls the science–game tension and highlights the unclear vision of CSGs. For CSGs to be widely accepted as enjoyable experiences, though, we agree with [DGL4]: the player experience must be enjoyable for the game to serve its purpose as a leisure activity that engages a wide audience.

Lastly, if you do make a successful game, [DGL4] warns against a common pitfall when publishing about it: don't let your research be simply "marketing talk" about how great your game is. Instead, we recommend to **(T16) focus publications on lessons learned and general takeaways that other projects can benefit from**. Contribute to the greater CSG community by open-sourcing your work, creating a platform for other CSGs, or providing other resources for getting started.

Finally, **(T17) help develop the CSG community** by collaborating with other CSG developers.

4.2.4 Discussion

In this study, we conducted a joint qualitative analysis of 57 interviews with stakeholder groups from 10 different CSGs to understand their differing individual perspectives and needs (Table C.1) as well as shared and/or cross-cutting challenges, namely the ambiguous allocation of roles, limited resources and funding dependencies, the need for a citizen science game community, and science–game tensions (Table 4.1). Here, we connect these findings to prior work and derive recommendations — both from our research and directly from our participants — on how identified issues can be addressed (Table 4.2).

The issue of ambiguously allocated roles mirrors findings in Science and Technology Studies (e.g., [50]) and by Wudarczyk et al. [569] that interdisciplinary teams need alignment on project expectations, a common goal, an understanding of different practices, agreement on terminology, establishment of shared knowledge, transfer of essential technical knowledge, and embracing diversity as an asset.

Counter to Golumbic et al., who found that scientists enter citizen science for funding reasons [186], we found that there is very little funding available — relative to their needs — for CSGs. This may be partially due to a 'cooling' of funder enthusiasm in the past 5+ years, as CSGs have become less novel. The 'lumpy' grant-based model of CSG development creates challenges for ongoing basic operations of game development, such as bug fixes and code refactoring, tutorial development, playtesting, and community development. Consequently, developers volunteer their time to these basic operations, leading to burnout and churn, which in turn exacerbates the problem, for example, as many developers enter the project briefly then leave, creating a messier codebase.

A third theme was a need for a CSG community. Several problems with CSG development currently could be mitigated with more shared knowledge, code, and other reusable resources. Despite the existence of communities for citizen science and for games, there is no community for citizen science games. This fragmentation means that most CSGs are starting from nothing every time. There is still no general solution for developing CSGs — most projects develop and grow dynamically, despite platforms such as SciStarter[2] to promote and support projects. As one possible first step toward a global CSG community, the authors have put together a Google group for CSG developers,[3] with 55 members at the time of conducting this study. We encourage anyone interested in CSG development to join this group and collaborate toward other ways of centralizing the community.

Fourth, we expand on prior observations of the science–game tension in CSG [392, 332, 394]. We break this tension down into several aspects: CSGs are developed as scientific software, development focuses on the possible not the polish, there are communication breakdowns between scientists and game designers, and implementing scientific tasks into gameplay is inherently difficult.

Notably, these four challenges interact and compound. Implementing a scientific task into gameplay may be hard, but it's even harder when there is no dedicated game designer on the team and developers are ambiguously in charge of design and development. These ambiguous roles are strained more by not being funded to work on them — with no funding to guide their efforts, they focus on what interests them, resulting in half-finished projects abandoned when they burnout and leave. Their absence impacts the player community, who gains distrust for new developers coming in, questioning whether they will stay and help or create a mess and leave as well.

Communication especially is an intersecting factor across many themes and individual challenges. While we focused on communication only with respect to the science–game tension, in many ways, communication is the core problem of ambiguous roles, building a CSG community, and building trust and competence with the CSG players.

How should these issues be resolved? Is it really true that science and gaming are epistemically opposed? We argue that the issues with CSG development are not unsolvable. Above, we

[2] `https://scistarter.org/`
[3] `https://groups.google.com/g/csg-developers`

listed recommendations gathered from our interviewees and synthesized from our analysis. While these recommendations are not a panacea for the problems in CSGs, we hope that this can be the beginning of a conversation which recognizes these problems and actively works to address them — together, as a global CSG community.

4.2.4.1 Limitations and Future Work

As stated previously, although this work set out to explore stakeholder perspectives on CSGs, we did not analyze all potential stakeholder groups, excluding funders, policymakers, and companies in the broader supply chain of CSGs. This was both because these stakeholders do not directly contribute to the production or consumption of the CSG player experience, but also a decision of scope for the purpose of this project. Therefore, future work could examine these perspectives.

Second, while we thematize our participants' descriptions of their experiences to the best that we understand them, it remains possible that we have missed or misinterpreted some aspects of their perspectives and our representations must always remain interpretations. Future research can validate this study through additional analyses, longitudinal studies, or other methodologies, as well as empirically testing the recommendations provided for efficacy.

4.3 Conclusion

This chapter summarized two studies which analyzed the experiences of players and other stakeholders of CSGs. The Player Study revealed five directions for improving CSGs: scientific communication, instructional design, interface and controls, task quality, and software issues. More specifically, the recommendations that precipitated from this study are:

1. Teach the core gameplay loop and scientific contribution model early.
2. Iteratively refine instructions and communication methods.
3. Improve the speed, clarity, frequency, and regularity of scientific communication.

In the Stakeholders Study, my co-authors and I found that the diverse stakeholder groups of CSGs, including project leads, scientists, developers, and educators, have diverging needs and tensions between them that need to be addressed for CSGs to realize their potential value. We generated several themes that represented issues across multiple stakeholder groups, including the ambiguity of roles, limited resources and funding, and tensions between the fields of game and science. We also list 17 concrete recommendations in Table 4.2 for meeting the needs and overcoming the challenges of each stakeholder group.

4.3.1 Takeaways

Do these perspectives tell us anything about the onboarding issues in CSGs? Quite a bit, actually. The problem, though, is that the challenges in onboarding are tangled in a messy mangle of play and practice [488, 385]. The needs of developers and users are caught in a complex tension of interdisciplinary and cross-disciplinary challenges. The players might need clear gameplay to engage and understand the task, but there is no dedicated game designer on the team because roles are ambiguously allocated. The developer who is responsible for gameplay might be pressured by funding to work on something else, or oppositely have no pressure at all and thus work on anything they desire, which may or may not be the gameplay that players wanted. The developer, likely a student, might burn out from volunteering their time, leaving a half-finished project to slow down future developments and giving the player community less reason to trust the development team.

We may not be able to solve all of the systemic issues of CSGs all at once, so instead I focus on what is more controllable: the designed onboarding. In the next chapter, I look at the specific issues players are experiencing with respect to gaining expertise in ECCSGs. In combination with the insights from this chapter, these identified issues should help develop strategies for better onboarding design in Part III.

Chapter 5

Barriers to Expertise

. In the previous chapter, I examined the general state of CSGs and the experiences of players. But what, specifically, is preventing players from learning how to play? What are the barriers to expertise, what are the challenges that players face when they are onboarded into one of these CSGs?

This is answered by the next study, which tries to identify the "breakdowns" [228, 227] in CSG onboarding, defined by Sharples as "observable critical incidents where a learner is struggling with the technology, asking for help, or appears to be labouring under a clear misunderstanding" [472, p. 10]. While some breakdowns can lead to learning, others can lead to catastrophic disengagement [228]. Therefore, a better understanding of which kinds of breakdowns are occurring in ECCSG onboarding — and how — is important to supporting players' expertise development for solving scientific challenges. More broadly, by understanding expertise development we may be able to gain new insights into how to train learners in new domains for which little to no training materials have yet been developed.

In this study, my co-authors and I examine the research question "What is the path to expertise in ECCSGs and what are the major barriers along that path?" This is achieved through interviews with ECCSG players of *Foldit*, *Eterna*, and *Eyewire* about their skills and experiences, and we explore the data using a deductive and constructionist application of reflexive thematic analysis [58, 59]. In doing so, we attempt to identify the golden path[1] to expertise in ECCSGs and any barriers along that path. More generally, the study of ECCSGs provides key insights into the design of game-based learning, crowdsourcing, and community knowledge construction. Although we list most recommendations in Table 5.1 as specific to ECCSGs, many of the principles can serve

[‡]Parts of this chapter were adapted from [332]

[1]In the games industry (and other user-centric industries), a *golden path* is the sequencing of activities leading to the perceived ideal experience or optimal outcome [525].

as inspiration for educational games, serious games, and other citizen science projects.

The primary contribution of this study is a model of the path to expertise in ECCSGs and barriers along that path. We find that the path to expertise is a cycle of exploratory learning and social learning. The three dominant barriers to this cycle are missing instruction, missing polish, and missing communication. Specifically, new players struggle to understand how to interact with the game and the broader, holistic research loop of how gameplay and science connect. Then, frictions with the user interface, technology, and gameplay slow or hinder exploratory learning. And finally, several communication barriers prevent the social learning that would otherwise ameliorate these other failings, including inaccessible and infrequent scientific communication, and the gatekeeping of community content creation.

Based on these findings, we provide recommendations for CSG developers, for example collaborating with professionals of complementary skill sets in community management, UI/UX, software development, game design, instructional design, and science journalism. We further recommend providing social features in and outside of the game, teaching the big picture first, and improving scientific communication with the players, among other recommendations listed in Table 5.1. However, in discussing these potential or partial solutions, we also note broader complications with the ECCSG model, such as their lack of financial sustainability and the accessibility issues to participation brought on by requiring expertise.

5.1 Background

Recall from Chapter 2 how research has been done on how players engage with games [142, 74, 452, 499, 215, 558] and what makes games playable [386, 129, 374, 276]. Within the realm of player experiences, the field of GBL focuses specifically on how players learn from games — often regarding games as constructionist learning environments [133, 180]. As instructional tools, games offer experiential learning via a range of mechanisms including active and discovery learning, forming affinity groups, cycles of expertise, well-ordered problems, and simplifying conditions [175, 178, 180, 426].

Yet, there are still times when players don't connect with a game, when the game-user interaction breaks down. Iacovides et al. [227, 228] explored this phenomenon and identified three types of breakdowns: Action (failure to execute an in-game action), Understanding (failure to figure out what to do), and Involvement (failure to engage, such as from boredom or frustration). They further found that macro-level expectations of the game were informed by prior experience, other players, and the wider community. Moreover, continued involvement (at the micro-level) depends on meeting expectations, both internal to the game (such as in-game rewards) and externally (such as the price to purchase the game). Lastly, a key factor for involvement is the experience of agency, which is reduced if players do not believe that they have a meaningful impact within the game world. So players can experience breakdowns when their expectations are not met or they don't perceive their agentic influence on the game.

Zooming back out, breakdowns are one type of friction along a player's path to expertise in a game. Expertise, in short, occurs when an individual "chunks" their knowledge into cognitive schemata in order to process information quicker and easier [460, 411, 293]. This has the added effect of experts seeing and representing problems differently from novices, because experts rely on structural rather than surface features for problem-solving [78, 73]. Within games, researchers have studied expertise in depth, from what players learn [158, 141] to how they learn [213, 223, 440] to how to figure out what they learn [511]. Yet, ECCSGs are a separate domain entirely and an interesting but unexplored niche for understanding expertise in games. ECCSG players have different (or rather, additional) motivations compared to commercial game players, and the element of scientific knowledge expertise in ECCSG play adds an unknown variable into how expertise is acquired compared to other player expertise.

Much has been written on the motivations of CSG players (see Section 2.1.2), but studies have also been done to understand the ways in which participants engage with non-game citizen science projects. Scholars identified five profiles of engagement: loyal, hardworking, persistent, lurking, and visitors [21, 391]. We expect that only the first three forms of participation gain expertise; however, lowering the barriers described in this work may also serve to admit lurkers and visitors into further engagement.

Synthesizing the factors of motivation, Jennett et al. developed the Motivation-Learning-Creativity (MLC) model [241]. Their model describes citizen science participation as initiated by a motivating interest in science. Then participants learn via participating at the micro- and macro-levels. Next, participants identify as a member of the community, which finally leads to creative contributions. Both identity and creativity then reinforce the motivation to participate. As shown in this study as well, Jennett and her colleagues highlight the importance of social learning, community building, and sharing. Participant learning is notably achieved through contributing, social interaction, using external resources and project documentation, and sharing personal creations, and as a result participants gain several learning outcomes including more knowledge on the scientific topic as well as scientific literacy. The MLC model is echoed in our findings as we identify expertise as a function of participation and social learning. Notably, the creativity component of MLC includes personal creations such as developing helper tools and resources and discussing ideas. However, we describe in Section 5.3.4 how there are barriers specific to ECCSGs which prevent this creativity from flourishing.

Despite the great amount of research on motivating participants through CSGs, only in the last few years has the CSG player's experience really been scrutinized. Ponti et al. [392] comparatively explored the players' experiences in *Galaxy Zoo* and *Foldit* and found several key themes, including: tensions between knowledge production and competition, frustrations as a result of gaming mechanics or practices, and questioning project goals. Eveleigh et al. [153] investigated the role of 'dabblers,' or casual contributors, in CSGs. Their findings supported Haythornthwaite's theory that intrinsically motivated volunteers are more likely to contribute in depth and form a community [209]. Moreover, their work highlights the importance of understanding and breaking down barriers to initial participation, such as by acknowledging contribution efforts, decreasing boredom, and enabling the players to fit the game around their existing schedules.

Díaz et al. [136] asked players directly about their game experiences. They found that players struggled to understand the game (*Quantum Moves (QM)* [296, 242]) and wanted better tutorials; as one player put it, "Both too simple tutorials and challenging game, too steep learning curve." Another expressed a desire for more scientific clarity, "Explain how the game works, make a link with the part of physics which it concerns, it was all a bit unclear what [it] is really all about. It worked for me but have not a clue what you accomplished with all data that is gathered. The idea to turn to the public is great, but explain more." The researchers concluded that providing tutorials could equip players with a better understanding of the game mechanics and increase participation and game interest. This desire for better tutorials was also found in a systematic literature review of citizen science volunteers more generally [477]. Volunteers asked for better tutorials, claiming that the help page and tutorials were among the least useful and least usable features (specifically, for *iNaturalist*), and several articles in the review discuss the need for providing tutorials in various forms.

Therefore, while we understand what CSGs are, why they're useful, what motivates players to play them, how the games themselves are experienced, and broadly how expertise is gained in games, we don't yet know how expertise is gained in CSGs and, in particular, in ECCSGs where expertise is a crucial factor of the game's success and its scientific contributions. The players' journey to expertise is also an important part of the accessibility and inclusion of CSGs. We already know from previous studies that CSGs have participation biases based on age, gender, and scientific capital — the older Western males already rooted in scientific culture have much greater access to making and benefiting from citizen science contributions [484]. If ECCSGs were to introduce an additional cognitive constraint (which they do), accessibility and inclusion is further restricted. This work is not just about making learning easier, it's about the public's right to participate in the production and consumption of scientific knowledge.

In summary, I see a gap in the literature in the systemic ways players are hindered or prevented from gaining expertise in ECCSGs, which is critical to the value of ECCSGs to make scientific contributions and engage the public in the production, organization, and circulation of scientific knowledge. This work seeks to empower CSG developers and their players to identify and overcome these barriers, lest the frustrations of expertise-centric CSGs disillusion the public and shy them away from a valuable form of citizen science. However, let us take a brief tangent to critically examine this gap.

Why is this gap appearing now, or perceived to appear now? First, the ECCSG model is relatively new, being first described by Keep in 2018 [255], so the concept is novel. Second, CSGs are generally a niche subject, so not many researchers are aware of their existence. This explains why expertise in ECCSGs have not yet been studied, but it doesn't explain whether the path to expertise in ECCSGs is *worth* studying.

First, as mentioned in Chapter 1, ECCSGs are the most difficult games to learn (that I know of). Therefore, a better understanding of how to teach them can lead to insights in teaching difficult concepts, which may provide transferable benefits to other education research. Second, as I argued in Section 4.2, ECCSGs are a complex, interdisciplinary, and cross-disciplinary assemblage of socio-technical systems and actors [406]. In this way, ECCSGs are a microcosm of some of the kinds of difficult problems and "grand challenges" we are facing and will face in the twenty-first century: interdisciplinary development, coordinating with large crowds of problem-solvers, and the generation, organization, and propagation of knowledge [253]. By understanding the operation and expertise development in ECCSGs, we prepare to understand larger and even more complicated networks of people and information technology. So, while this "gap" might not be a necessary concern, the act of "filling" it is a step toward better education and the organization of solving complex problems.

5.2 Methods

Methods were approved by the institutional review board at the authors' institution. The three games examined were *Foldit* [99], *Eterna* [287], and *Eyewire* [514] (see Appendix A for details of each game). As shown earlier in Table 2.1, there are only a few games (to our knowledge) currently available meeting enough criteria to be considered ECCSGs. For the purpose of this study, we focused on ECCSGs that produce expertise concretely, since the organization of knowledge is a key component to understanding expertise-formation. This inclusion criteria helped us become familiar with the games studied and framed experts' input in a larger body of knowledge, allowing us to better analyze and thematize results. We also excluded games that are no longer available, namely *Decodoku*. A study of games with less concretely published expertise (*Mozak*, *QM*, and potentially other games we are unaware of) is left as future work.

5.2.1 Participants and Protocol

Purposive sampling was used to recruit ECCSG players (n=16: 12 *Foldit*; 3 *Eterna*; 1 *Eyewire*) — via game website messaging systems — from diverse backgrounds of expertise ranging from very novice to extremely expert. This size was found to be pragmatically sufficient (cf. "saturated" [197, 60]), pragmatism being a recent heuristic to address the problems with defining theoretical saturation as "no new information" from a qualitative analysis; instead, we let go of the notion that research would ever lead to a definitive stopping point and instead stop data collection based on practical constraints while still ensuring our analysis forms a coherent conceptual model [60, 305].

The skew toward *Foldit* exists for three reasons: first, this sample is proportional to the community size for each game — *Foldit*'s active player base is several times that of *Eterna*'s and *Eyewire*'s.[2] Second, the authors — being *Foldit* developers ourselves — have more experience with *Foldit* and can better analyze player descriptions of expertise. Third, prior literature has more thoroughly investigated *Foldit* (as a game), so there are more points of reference for comparison (e.g., [114, 98, 392, 395, 159, 257, 3]). Moreover, our analysis focused only on phenomena represented across all games, with the exception of subsection 5.3.4.3 which has been found previously by Ponti et al. [392] and is described here for the purpose of connecting their result to a broader dilemma in ECCSGs.

[2]Estimated based on media reports of registered players [56, 533, 505], game community activity (e.g., Discord), and personal experiences in each game's chat room during the time of writing.

No other demographic data (e.g., age, gender) were collected. This was both to protect our participants' anonymity and because expertise is not causally linked with these variables. However, some of our participants have publicly released information about themselves via in-game profile pages, and based on these data we believe our sample to be an accurate representation of typical CSG populations [115]. Further, we cannot provide clear measures of expertise because each player's experiences vary. For example, some players have played on and off for several years and were unable to recall exactly how much experience they have, in terms of months' experience or hours played. Instead, we allowed players to self-report expertise and collected experience reports where available. We found that players generally self-reported as novices at less than 2 years of experience, intermediate at 2-5 years of experience, and expert at more than five years of experience. Therefore, we use this heuristic combined with self-reports when describing players quoted in this study. This categorization resulted in 3 novices, 9 intermediates, and 4 experts.

Participants were interviewed online for about an hour in a semi-structured format about their play experiences, their skills in the game, and how they conceptualize those skills. For example, some questions asked were "What is the very first skill a player needs to learn in [game]?" and "How is the process of becoming skilled in [game] similar and different to becoming skilled in [other games or other hobbies they feel skilled in]?" Other questions included asking about the visual cues for expertise, how the player would hypothetically redo the tutorials with an infinite budget, what they wish they knew when they first started playing, etc. Participants were then offered a $15 USD Amazon gift card as remuneration. The interviews were audio-recorded and then transcribed for further analysis. In total, 16.2 hours (per participant: M=60.75 minutes, SD=11.86) of data were collected.

5.2.2 Analysis

I analyzed the data using reflexive thematic analysis in order to take a "Big Q" [262] qualitative approach to our research question [58, 59]. The use of a fully qualitative research method is required here because: (1) our target population is small, so quantitative methods would be impractical; (2) positivist measures of expertise across a range of domains (different games) would struggle to make meaningful comparisons between domains; and (3) the nature of our research topic (gameplay and player experience) is strongly subjective. Moreover, we were not interested in a close analysis of language use (which would suggest interpretive phenomenological analysis or discourse analysis) or a post-positivist content analysis (and related "little q" codebook approaches), and our sample is too small and homogeneous for grounded theory, making reflexive thematic analysis the optimal choice for our goals and constraints [61].

The analysis approach was primarily deductive, latent, and constructionist. We theoretically grounded the analysis in constructivist and constructionist theories of learning and play (i.e., games as constructionist learning environments [133], play as constructing predictive mental models [15], and games as constructivist affinity spaces [180, 460]) and aimed to answer our research question: "What is the path to expertise in ECCSGs and what are the major barriers along that path?" We additionally took a critical orientation to sense-making, but we included an element of critical realism in that we were open to the data providing evidence against our assumption that player experiences are explainable by our theoretical framework. Therefore, we code both for semantic and latent meaning in order to capture the overt player experiences as well as how these experiences might be interpreted through our theoretical lens.

The analysis occurred in six rounds of iteratively passing through the data to apply codes, merge codes into themes, and return to the data to validate and refine themes. As stated above, codes were both semantic (e.g., "uses Wikipedia") and latent (e.g., "exploratory learning"). During the initial coding, all codes were unique and descriptive. Subsequent rounds of coding then oscillated between aggregating codes by similarity and verifying the new codes were still accurate to the original transcription. By the fourth iteration, codes were aggregated enough to be representable as themes which were then refined over two additional rounds of analysis. In order to minimize the effect of the skew toward *Foldit*'s population, special attention was given to ensure that themes were grounded in data from participants of all games and not found only from *Foldit* players. Note also that the analyst is himself a *Foldit* developer; although this imparts a kind of bias to the research, it also uniquely positions us to understand the situation from both the player and developer perspectives. And, as stated earlier, emphasis was placed on ensuring themes generated were evidenced across all games to reduce bias toward *Foldit*. For validation of our methodology and future research, an audit trail and additional quotations are provided at: `https://osf.io/hn7x2/`.[3]

Finally, after themes had been generated, the participants were consulted again for transparency — to verify that the results below accurately represent their beliefs and experiences. This check was performed in case the researchers had misunderstood or misquoted the participant, or changed their meaning by taking a quote out of context; this resulted in one quote being clarified but did not affect the themes generated.

[3] For privacy, please contact me directly regarding access to anonymized transcripts.

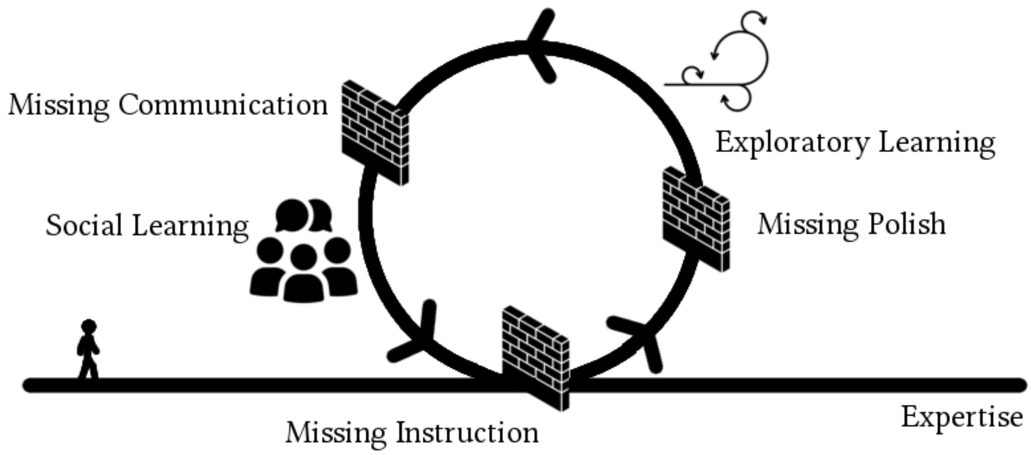


Figure 5.1: A visual representation of the main themes generated from the reflexive thematic analysis. The path to expertise is a cycle of exploratory and social learning with three barriers: Missing Instruction, Missing Polish, and Missing Communication.
Icons licensed under Creative Commons from the Noun Project: Social by Adrien Coquet; iterate by Justin Blake; brick wall by Bakunetsu Kaito.

5.3 Results

The results of the analysis are shown visually in Figure 5.1. The path to expertise was found to be a cycle of exploratory learning followed by social learning with three major barriers along that route: onboarding and continual learning carry the instructional barrier of Missing Instruction, then exploratory learning is hindered by a host of game-related barriers wherein Missing Polish causes friction with the game interactions, and finally social learning is blocked by Missing Communication, the sociocultural barriers. The remainder of this section unpacks each of the four themes (path and three barriers) in subsections.

5.3.1 The Path to Expertise is Social and Exploratory

One major question of this research was "What is the path to expertise in ECCSGs?" This question is perhaps answered most succinctly by P11:

> *It's a combination of, like, messing around and then looking it up somewhere else and asking somebody.*[4] (P11, *Foldit*, Intermediate)

[4] Some quotations are abridged for readability.

In this way, player learning is both constructionist and constructivist. They first explore, constructing their own mental models, and then exchange tangible artifacts (i.e., puzzles and solutions) and ideas as a community, building a collective knowledge base which iteratively informs further exploration. Notably, as shown in Figure 5.1, the order is important. Players highlight that in this cycle of explore-discuss, experimentation comes before learning in order for the problem space to become meaningful:

> *I have to struggle with it before it has any meaning to me to look it up online. Like I could just look it up online, go, oh, look at the perfect thing. But it doesn't have as much meaning to me unless I sit there and poke at it for half an hour and then look it up.* (P5, *Foldit*, Novice)

Compare this to Schwartz and Bransford's "time for telling" [461], the notion that students require the time and space to discover a problem and its significance before being told the solution(s). In line with constructivist learning, players are grappling with a problem before being ready to understand ways to solve it. They are experimenting and engaging in trial-and-error discovery learning. P9 (*Foldit*, Expert) describes she would "consciously set goals for myself... I was actively experimenting with stuff... I was ... actively chasing this knowledge."

However, the "time for telling" approach works best as guided discovery or scaffolded inquiry, rather than unconstrained discovery [461, 63, 87]. What the players described was very much unconstrained, unguided, and unscaffolded, suggesting that the value of experiential learning may be an artifact of the currently available guidance for players. "Most of the expertise that you need to operate the game do not appear in the tutorial," says P13 (*Eyewire*, Intermediate). "They are gained from experience and trial-and-error."

In this way, the game effectively forces players to learn by experimentation because no other option is provided. If the game were better scaffolded through carefully crafted instructional design (e.g., following Quintana et al.'s scaffolding framework [413]), would this learning path still be relevant? *Minecraft* [341], for example, began with a similarly unguided social-exploratory onboarding experience, but as its *Education Edition* has grown in popularity, there are now over 600 guided lesson plans for teaching *Minecraft* and teaching with *Minecraft* [279, 32].

Complementing the exploratory, the other half of the path to expertise was social. Across all three games, players consistently emphasized the role of social learning as deeply embedded in their path to expertise. P3 (*Foldit*, Expert) emphasized socialization, team connections, and community as fundamental to his experiences. As one example, he praises the asynchronous cooperation of being able to "hand off" his work in the evening to a teammate halfway around the world who was just waking up and could continue his efforts. Additionally, the community acts as a collective source of knowledge and motivation. "There's a community of veterans that you're stepping into who can share information and encouragement," says P9 (*Foldit*, Expert).

This constructivist learning is bolstered by several forms of peer learning that the players described, including peer modeling, peer tutoring, and peer assessment [185, 521]. Through this social engagement, players form lasting relationships with each other that extend beyond aspects of teaching and learning:

> *[The process of gaining expertise] more and more relates to how it connects to players that eventually helped. And what's most important to me was the people I eventually found that, you know, that you build a relationship with and you build trust in and you really respect around the issues of science.* (P14, *Eterna*, Intermediate)

To use the language of Gee's Discourse analysis [176], these tight relationships eventually become portals to the semiotic social space or affinity space [179] — in other words, player relationships create entry points into connecting with scientific issues and assimilating into such a community. While the game generates the explorable space, veteran players act as portals for new players to enter and assimilate into that space.

To summarize, the path to expertise is currently a cycle of experimenting and discovering via social learning and the use of paratexts [94] such as guides and wikis to "look it up" (P5 and P11). There are several noteworthy comparisons between this and Jennett et al.'s MLC model of engagement and participation [241].

Jennett et al. use the language of "micro" and "macro" tasks (calling back the Gaming Involvement and Informal Learning framework [230] and the Player Involvement Model [68]) to speak of the in-game, moment-to-moment play as the micro-involvement and the external, surrounding, or off-line activities as the macro-involvement. Using this language, then, we claim that expertise is a cycle of micro and macro involvement in that order: interactions with the game itself (Gee's "generator" [179]) triggers interest in socializing and the use of paratexts external to the game ("portals"). Engaging externally then informs later play, giving the player newfound language and ideas for interaction, and thus placing the media object itself at the center of this "big D" Discourse [181]. Jennett et al. observed this learning as well, though not with the sequential distinction we place on it now: they found that participants learn by the micro-involvement of contributing and the macro-involvements of interacting with others, using external resources and project documentation (paratexts), and sharing personal creations (such as creating their own guides) [241].

However, neither halves of this cycle speak to *why* players engage with the game. Although — as noted in Section 2.1.2 — much is known about CSG motivations, we sought to verify if these motivations hold true in the context of expertise-centric play. Based on player input, we argue that engagement has dual drivers: the scientific value and self-gain. This expands on previous work, which describes the motivations of "altruistic factors" (e.g., [114, 414]) or "personally-focused reasons" (e.g. [20, 437]) [21, p. 247], to specify that something from each of these motivation categories is required for continued play.

Some players began play for the scientific contributions; others began for themselves, such as for entertaining gameplay or personal learning. But for those who have reached expertise, it becomes clear that both motivations are required for expertise: one needs to be invested both in the science and their own benefit to continue playing:

> *The chance to both do hard puzzles and contribute to science. It's really what got me interested and what has kept me interested.* (P9, *Foldit*, Expert)

Although not a requirement, the most common form of self-gain was intellectual challenge:

> *I've been looking for something intellectually challenging to get involved with that I find interesting... I have a pretty strong science background from college that I never really used but I'm a bit of a science geek. So that was very exciting: the idea of designing RNAs and having them synthesized and tested in a lab was very exciting.* (P16, *Eterna*, Intermediate)

Most likely, the skew toward intellectual challenge is because all existing ECCSGs fall within the puzzle genre. This may not generalize, since there is nothing in the definition of ECCSGs which requires them to be puzzle games. This finding is consistent with previous research on the motivations in CSGs which state that players are drawn into CSGs by their previous interests in science, the specific research topic, curiosity, and a desire to contribute to research, and their continued engagement is dependent on intrinsic enjoyment, proper pacing, teamwork, community, and intellectual challenge [241, 114, 229, 513].

Another key aspect in the path to expertise is the development of what Goodwin refers to as a "professional vision" [189]. This is not a new finding, as Ponti et al. [395] discovered this several years ago in *Foldit*. However, the present study confirms that expert players of all three games studied describe recognition, both of the problem space and solution space, as a critical skill of expertise and a factor of decision-making. Some *Foldit* players describe this professional vision as an aesthetic intuition:

> *I call it a protein aesthetic. I think after you play with it for a long time, you get a sense of what looks good. Y'know when we chat with each other in group, we'll say "that looks beautiful." And what we mean is it conforms to how we have come [to know how] proteins [look] when they're correctly folded.* (P3, *Foldit*, Expert)

In fact, this professional vision is so ingrained, so intuitive, that participants struggled to talk about their expertise and learning process, which is perhaps an obvious finding for researchers familiar with CTA. As CTA researcher Clark writes, "experts don't know what [others] don't know," in other words, experts significantly distort or omit details of their own expertise without specialized knowledge probing [84]. P12 (*Foldit*, Intermediate) says that they "can't really put into words what makes a well-designed protein." P7 (*Foldit*, Intermediate) describes it as feeling like he is a neural network and not being able to describe his own weights and biases: "It's intuition for me. It's more just like deep understanding... I don't know, I'm describing how to, to do 2+2 is 4, it's strange."

Novice ECCSG players also recognize that this intuition is one of the core elements of expertise in the game:

> *The peak performance play is when you know that something is in the wrong place or in the wrong shape and you can try to impose your will into the game.* (P8, *Foldit*, Novice)

Moreover, our study confirms that this phenomenon extends beyond *Foldit*. *Eyewire* and *Eterna* players also described having a professional vision, using much the same language as *Foldit* players. P13 (*Eyewire*, Intermediate) says that working with difficult data in *Eyewire* involves "knowing how a cell is supposed to look."

Similar to other domains of expertise, we observe that ECCSG experts "see" the domain space differently, using unconscious structural heuristics to guide problem-solving [73, 78]. Identifying this professional vision is critical in light of Keep's philosophy on ECCSGs: because the value of these games is the expertise itself, rather than the problems solved by expert players, it is absolutely necessary that developers provide support for documenting, sharing, and organizing their expertise [255]. Without developer assistance, players will struggle to articulate their own learning and knowledge. Even in successful examples of social learning in games, much of the knowledge is "impenetrable" without first spending time submerging oneself in the game experience and the online community [486].

5.3.1.1 The Role of Science Knowledge

Because ECCSGs involve scientific domain knowledge, we also sought to explore how knowledge of the relevant scientific topic connects with expertise in the game. Is it a barrier to expertise, a necessary component on the path to expertise, or something entirely tangential and parallel to game expertise? Interestingly, we found that the role of science knowledge was different for each participant. This may signal that there is another theme to be explored here as future work. Given our sparsity of data, we draw no substantial conclusions, but offer several perspectives. These perspectives can be seen as forming a spectrum, from science having no connection to the game to the two being integrally linked. The first perspective, offered in third-person, is that the science is irrelevant, most players simply want to play a game:

> *Our current two labs we have going on are very science heavy and I think there's some players who don't like that. I think the majority of players, they like having that wall there. "Just give me a game to play." And... They want us, they want the advanced players and the developers to create the definitions, to understand the science and create the definitions and say "here, we need you to solve this as a puzzle." And, having too much science involved, they don't want to read it. They don't want to read it. They come on to have fun for an hour. They don't want to be reading about science, they don't want to be reading. So I think there's an ongoing debate about how much science to provide to most players, whether they want it or not. Whether they* need *it or not. And I don't know what direction it's going to go in and which is best.* (P16, *Eterna*, Intermediate)

The second perspective is that the science is important, but actively learning the science is not necessary because it's intrinsically embedded into the gameplay:

> *[Learning the science] kind of just happens naturally because the metagame in Foldit is inextricably linked to the actual physical science of good protein folding.* (P1, *Foldit*, Intermediate)

The third perspective is that you don't need to learn the science to play, but you do need to learn it to play well:

> *It was said that basically you don't have to know anything about organic chemistry or, you know, or molecular chemistry... in order to do these. And my answer to that would be, yes, that's true. But if you want to do those puzzles and do well at those puzzles, I believe that you really have to know something. I mean, to me, it's like, sort of like playing chess. You know, anyone can learn the rules and move the pieces [but] to be an expert chess player, you know, it's going to take years and years of studying. And so I feel honestly that a person who has taken chemistry courses or biology courses or whatever, you know, I think that they're probably in a much better position to do those puzzles.* (P6, *Foldit*, Novice)

Finally, the fourth perspective is that the science is critical to motivating play, and in fact the most difficult part of onboarding:

> *I think I struggled the most with my own knowledge and to get the knowledge I have now, this was the hardest part, I think. And if I wouldn't be interested in protein design or biochemistry, I don't know if I would have kept playing... Be interested and get the knowledge, that was the hardest part, I think.* (P10, *Foldit*, Intermediate)

Despite this open question of whether all players need to learn the science, it seems that at least some players need to know the science so that there can be a bridge in communication between scientists and players. P16 (*Eterna*, Intermediate) describes another player who "is invaluable because he is a player and he understands the player perspective, but he also understands a lot of the science, and the scientists, and how they think." Player-scientist communication will be further unpacked in Section 5.3.4 as participants describe how science jargon is one of the major barriers to understanding their contributions and how to play.

This finding is most easily interpreted through Harteveld's Triadic Game Design framework, which posits that serious games are a confluence of three interconnected components: reality (e.g., protein design for *Foldit*), meaning (in ECCSGs, learning and scientific contribution), and play [206]. The value of this model is that it allows us to view design issues as problems within each 'world' ("tensions") and at the intersections between them ("dilemmas"). Specifically, the role of scientific knowledge is offered here as a 'trilemma' of ECCSGs, a tension from the interplay of all three aspects of citizen science gaming. What role does science knowledge play within the game, and how can scientists and developers teach it? It is clear that the scientific topic is, in some way, connected to the gameplay. But is it a requirement, a distraction, or altogether parallel? The answer is most likely that it plays different roles for different players, depending on their prior knowledge and interests. And if so, how can scientists and developers handle the science differentially for each player? This paradigm would seem to require CSGs to both teach the science effectively and let it go ignored should the player choose. Currently, neither goal seems to be satisfied, given player frustrations from each side.

To summarize, we found that the path to expertise is: (1) built on constructionist (exploratory) and constructivist (social) learning, (2) requires both motivations from scientific contribution and self-gain, and (3) involves the development of a professional vision; points 1 and 2 are novel contributions of this study while point 3 is a confirmation and extension of previous work [395]. With the path to expertise defined, we generated three themes to identify the barriers along that path. The first barrier encountered is Missing Instruction as the players are onboarded. Then they experience the friction of Missing Polish while exploring the game and the barriers of Missing Communication while seeking social assistance.

5.3.2 Missing Instruction (The Instructional Barriers)

As described in the previous section, although the path to expertise was found to be social and exploratory, this may be because the games themselves were insufficient at providing instruction. Perhaps the largest barrier of all, participants described an incredibly steep learning curve for all three games, which is likely an important factor in player retention and churn. This steep learning curve comes in part from a lack of guidance (see also, unclear gameplay in Section 5.3.3):

> *You had no advice from any tips or stuff like this, how to start... with a long strand of amino acids? Yeah, just "Go for it. Build a protein." And I didn't know where to start, how to improve a structure, what was a stable structure for a given [sequence of] amino acids.* (P10, *Foldit*, Intermediate)

Not only is the learning curve steep, it also tends to be long. P2 (*Foldit*, Expert) claims the average duration required to understand *Foldit* is two years. If the learning curve in practice is anywhere close to this length, then learning how to play the game is creating a significant lag for scientific progress in ECCSGs. One *Eterna* player describes in detail how the tutorials were simultaneously too difficult and unhelpful — so much so that she felt at first as if the developers were intentionally trying to gate-keep players from contributing to the science of the game:

> *It was very difficult. There [were] a lot of concepts to learn, and I found I did best by doing an hour a day. I didn't know any of the concepts, so I had to learn it all and let it sink in. So I had to learn the concepts. It's a lot to learn to take in — and you have to learn 'em, there's no other way around it. You just have to learn 'em. And then several of the puzzles, the tutorial[s] were really too hard and there wasn't any benefit to 'em. They were, I felt they were more challenging the critical thinking skills and trying to see, well, are you actually smart enough to solve all these hard puzzles or else we're not going to let you in. So to me... they made it challenging on purpose to try to reach the people who were really smart. Some of the puzzles were really hard. It took a couple, several hours to figure out. And I think they're going to change that, so I don't think that's the intention... So I felt some of the puzzles were too hard for no reason. But. I learned what I needed to learn in the puzzle and the puzzle progression. But it took me [[pause]] 40 to 80 hours, somewhere in there, a* long *time, and there's very few people who would dedicate that kind of time to learn the concepts if we're playing a game. And I try to tell them that... They never really believe me. I think they think it's much faster, the progression, than it is.* (P16, *Eterna*, Intermediate)

The other point that P16 makes is that players expect the learning curve to be shorter. Indeed, if the road to making scientific contributions is gated with such a steep, long learning curve, then ECCSGs raise the question of who is allowed to contribute to and benefit from scientific knowledge production [255]. One case study on the citizen science project *Supernova Hunters*, for example, created a bias of participation based on when datasets were released for volunteer effort [484]. In this way, project logistics may accidentally create accessibility barriers for those with less time or availability to contribute. Similarly, the skill barrier of ECCSGs forces volunteers to deeply engage to be allowed to participate, despite the fact that most members of the public don't have the resources to commit to such a deep engagement [255]. Participation biases strongly toward older (68% over the age of 40) males (as much as 78% in *Foldit*) from Western countries, as found in a survey paper by Curtis [115]. She also notes that many citizen science projects are biased toward appealing to participants with more *science capital* [19], or cultural and social capital related to science, such as scientific literacy, consumption of science-related media, and more opportunities to engage with science culture. Recalling from Chapter 2, there are two ways to lower this skill barrier: either the projects must recruit only volunteers with existing expertise or make expertise accessible to all volunteers in a just and inclusive way (common in data-centric CSGs) [255].

In addition to the entry barrier, ECCSGs fail to provide feedback on the players' work and overall progress. This challenge was clear in all three games, which require creative and complex solutions but do little to help the player iterate and improve their work. "At some point... I just can't advance any further," says P6 (*Foldit*, Novice), "I just have no idea of what to do at this point."

Instructional design theory suggests that both cognitive and corrective feedback is critical to learning [539]. Moreover, nearly by definition games require feedback to complete the user-system interaction loop by providing a "quantifiable outcome" [445]. Although these games have some feedback, it is insufficient for the player to make meaningful behavioral adjustments, thus calling into question whether the player can be gamefully attached to the outcome at all [247]. As P1 (*Foldit*, Intermediate) describes, "*Foldit* will grade you quantitatively, but it won't grade you qualitatively. And that's huge. I think that's huge. You'd need other players to help you out there." Because the game fails to give feedback, players need to seek it out from each other. This can add social pressures, especially for novices who are not yet comfortable engaging with the community. P8 elaborates on this tension of wanting more thorough feedback but being afraid of judgment and criticism from the community:

> *...You don't want to, like, put yourself on blast [i.e., embarrass yourself] ... go on the Discord server and say, "hey look at this thing that I created." Where it's a lot easier to*

> *passively absorb [other community interactions and community content]. I just sort of observe what other people succeeded with and just try to passively copy that off rather than sort of opening yourself up to... "Well, no, that's not the best way to do it." ... You know, you have to sort to put yourself out there in order to receive feedback in the first place where you could just say, "well, I could just avoid that all together."* (P8, *Foldit*, Novice)

Not only is there a lack of feedback on the player's attempts at solving a puzzle, there is also a lack of feedback on what these solutions mean in the broader context of scientific contributions. P9 (*Foldit*, Expert) notes that one of the common questions they see from players is: "am I really contributing?" As will be elaborated in Section 5.3.3, this connects to an unclear and often opaque loop between gameplay and scientific progress. Because there is no in-game feedback on scientific contributions, and rarely any external feedback on broader scientific achievements, players have no indication as to whether they are helping, which is one of the critical motivators for retention [241]. The motivational impact of scientific feedback and recognition was also found by Eveleigh et al. [153]. As one of their participants describes, "I lost motivation to continue contributing information because I was not sure how useful my input was." Indeed, based on Juul's definition of a game [247], players must be attached to the outcome: are CSGs still a game if the outcome is never made known to the player?

This issue also speaks to the broader problem of instruction: a lack of clarity on the elements and goals of the game. Part of why the learning curve is so steep is that the tutorials fail to answer "What am I looking at?" and "Why am I doing this?".

> *I'm not sure how this simulation relates to actually putting stuff in a jar and adding whatever. Like I said, I don't quite know. I'm sure there's a whole complicated process where they can actually make these molecules in real life. But it seems very far away from the game.* (P5, *Foldit*, Novice)

Although these are citizen science games, the actual science is disconnected from the game itself. As noted in previous work on breakdowns, involvement in a game requires the player to believe they are making a meaningful impact on the game world [227, 228]. When applied to citizen science games, the scope of the game world extends beyond the game to a real scientific laboratory. We can thus infer that player involvement further requires the belief of making a meaningful real-world impact. A lack of feedback on what the player is doing and how it relates in-game score and progress to real-world progress is therefore detrimental to the player's sense of involvement. The game elements, without context, become extraneous:

> *When I started playing Foldit, I just saw a score, but I didn't know what the score was... In the tutorials you see, OK, if you turn this amino acid to the right you get a higher score. But why? This was not so clear for me.* (P10, *Foldit*, Intermediate)

When scientists *are* able to provide meaningful feedback, it is met with joy from the players. During the COVID-19 pandemic, *Foldit* released several puzzles related to the novel coronavirus and provided extra feedback on lab results regarding them. P9 was delighted by this and describes how she wishes scientific feedback always came that often:

> *I'm frustrated by how slow science is... [Interviewer: Slow in what way?] To hear any results. Every time [Foldit] posts a video or a blog or anything that says "These solutions look good, here's some problems with those." I* love *that. Like "Oh, finally. Thank you! I needed some feedback," you know, and I think that, I think the community would thrive on just as much of that as you guys would give us. The fact that the Coronavirus feedback is coming so soon after the puzzle closes is* fabulous. *I* love *that. I wish that would* always *happen.* (P9, *Foldit*, Expert)

In addition to a high entry barrier and lack of feedback, there is a throughline running across both of these: the instruction of these games is missing the bigger picture. The tutorials teach the "micro tasks" but leave unanswered questions of "what am I looking at? And what are the little goals that go with looking at those things?" As P9 (*Foldit*, Expert) explains, "When you're in the tutorials, you have no idea what the macro tasks are."

The utterance of a "macro task" suggests that the best theoretical lens for understanding this player's experience lies in instructional design, specifically Reigeluth's elaboration theory, which posits that teaching is more effective when focus is given to the high-level concept between periods of elaboration on sub-concepts [426]. Similarly, van Merriënboer's 4C/ID model [540, 539, 542, 538, 536, 537] gives focus to whole tasks, or the "macro" tasks that P9 describes. The 4C/ID model in particular is designed for complex learning, befitting ECCSG domains. Thus, according to P9, *Foldit* fails to stress the importance of whole tasks, teaching only basic controls through part-task practice and never orienting the player to the actual scientific challenges that they will be expected to complete. Indeed, whole-task-oriented approaches appear to be far more effective (for cognitively complex skills like ECCSGs) than the isolating alternative [298, 538].

Finally, for intermediate players the biggest barrier is that these games fail to teach key concepts required for expert play. While the tutorials may adequately cover some basic control schemes, they fail to introduce advanced concepts that are considered integral for actually solving scientifically meaningful challenges. Combined with the barriers to social learning discussed in Section 5.3.4, this results in significant difficulties during intermediate onboarding.

> *[The tutorials] are pretty [[pause]] light. They don't... go into any great detail. They pretty much show you what the basic controls do, and then throw you in the deep end.* (P13, *Eyewire*, Intermediate)

Earlier in conversation, P13 had explained that "Most of the skills that you actually use... the harder skills are not even attempted to be taught." Instead, he says, knowledge of these skills is gained through trial-and-error. P13's phrasing "then [they] throw you in the deep end" is also echoed by *Eterna* players, who note that there are no tutorials for the scientific challenges of the game. "The initial tutorial to learn basic skills to get access to the labs is quite good," says P16 (*Eterna*, Intermediate), "but after that, once you're in the labs there are no more tutorials."

On top of leaving out important concepts, the tutorials sometimes leave in extraneous concepts. P1 (*Foldit*, Intermediate) notes that *Foldit*'s tutorial "...misses the mark in a lot of ways, I think. It teaches a lot of different tools that aren't really used in the normal metagame. And then the things that you do need to know, you more or less have to figure out from other players." Again, across these statements, we see the thread of social and exploratory learning as a fallback when the games fail to provide adequate instruction. If we look for instructional principles which might explain these frustrations, we see that CLT identifies the extraneous concepts (tools not used in normal circumstances) left in tutorials as added cognitive load which increases mental effort and reduces learning efficiency [501, 249, 224].

Thus, ECCSG onboarding suffers from four types of missing instruction: a high entry skill barrier, a lack of sufficient feedback, a failure to explain the bigger picture, and a failure to teach all necessary concepts for intermediate to expert play. Although these problems are largely present for only the onboarding, the failure to teach intermediate and expert concepts remains an issue well into expert play, and the lack of sufficient feedback is a constant barrier throughout a player's journey to expertise. This is why the barrier of Missing Instruction has been placed on the on-ramp to the cycle of expertise in Figure 5.1: it is both an onboarding issue and a persistent barrier throughout the player's journey.

These failings are explainable by common instructional design principles [501, 426, 540]. As will be summarized in Table 5.1, ECCSG development teams can mitigate this barrier by collaborating with professional instructional designers, professional game designers, and expert players to provide effective and enjoyable tutorials that include in-depth feedback systems and a focus on the macro tasks, core gameplay loop, and contribution framework.

5.3.3 Missing Polish (The Game Barriers)

When it comes to promoting expertise, the game itself plays a key role as the central media artifact that players interact with. At the superficial level, players interact with the user interface (UI) and input control scheme. Beyond this gulf of execution is the gulf of evaluation [364], in this case the game design and gamification design of the task. Mediating this interaction on all levels is the technology, i.e., the game as software. Because expertise relies on exploratory learning, these levels of interaction are the interface between the player and their experiences in experimenting with the game. Alone, none of these levels are as obstructive as the instructional barrier; yet, each level of interaction is a frictive surface that slows, frustrates, and hinders learning and engagement.

The primary problems with the UI in current ECCSGs are a lack of discoverability and intuitive control. When combined with the instructional problems described above, this leads to an overwhelmingly difficult entry experience, as the tools one needs to play are hidden or difficult to navigate. And since viewing the problem is a critical first step to solving it [336], UI issues create downstream effects on solving the tasks at hand.

> *It is so hard to play Foldit if you don't know how to kind of manipulate what's in front of you. And if you lose the patience to do so, then that's hard.* (P1, *Foldit*, Intermediate)

Even compared to other molecular visualization software, *Foldit*'s interface is described as "maddening," requiring "a lot of trial-and-error and frustration" (P4, *Foldit*, Expert) to get comfortable with tasks that P4 already knew how to do in other software. Another player described it as "fighting the UI a lot" (P12, *Foldit*, Intermediate), speaking directly to this wide gulf of execution. Even in *Eyewire*, the most difficult task is manipulating the camera and correlating between the 2D and 3D views of the game (P13, *Eyewire*, Intermediate).

Beyond the UI, there were many problems identified with the games' designs and mechanics, to the point where some participants questioned whether they were playing a game at all or simply scientific software made to seem exciting. The issues that participants raised were clear violations of what are normally considered heuristics of good game design: non-intuitive gameplay, unclear goals and scoring, and poor tutorial and level design [127, 129]. For example, P8 was confused by the core gameplay loop, lacking an understanding of the basic premise of how one is supposed to interact with the game:

> *It doesn't tell you that that's how it should be done. So I'm not sure that's how I'm supposed to do it.* (P8, *Foldit*, Novice)

To recall again the work of Iacovides et al., this breakdown of macro-level understanding inhibits involvement by not meeting the expectations of gameplay [227, 228]. In fact, the players have very few expectations. Macro-level expectations are informed by prior experience, other players, and the wider community [228]. Yet, new players often have no prior experiences with the niche of CSGs, and — as detailed below in Section 5.3.4 — there exist barriers to social onboarding, which includes expectation-setting. In this way, players have little precedent and preparation for what to expect from the game, and this creates friction when the game itself adds no further explanation.

However, a more common friction was participants describing a lack of reliability with the gameplay experience. For *Foldit*, this took the form of "finicky" tutorial levels:

> *It was like, I moved a sidechain and I wiggled it and I got the puzzle. I have no idea how. And you try the exact same thing. It didn't work the next time. You have no idea how it happened.* (P12, *Foldit*, Intermediate)

For *Foldit* players, whether they win a level or not with a given strategy seems up to chance; the game gives no feedback on how effective their strategy is or how they should improve, creating breakdowns of both action and understanding without opportunities for breakthroughs that might engage the player [228]. For *Eyewire*, this lack of reliability takes the form of skewed scoring in the gamification system. P13 describes how the AI agent which assigns tasks can make mistakes that penalize the player:

> *When I play Eyewire, mistakes made by other players and the game's level generator sometimes negatively impact my score. That is pretty discouraging. So I think the penalty and the scoring system could use quite a bit of improvement. [The participant later clarified that the game also severely and disproportionately punishes the player for "their own silly mistakes."]* (P13, *Eyewire*, Intermediate)

So not only do players struggle to interact with the game because of unintuitive controls, undiscoverable interfaces, and unclear gameplay rules, but when they are able to perform an action, the results are often unpredictable. This violates standard usability heuristics, such as Nielsen's principles [360] and the System Usability Scale [62, 31].

These issues are worsened by several technical difficulties that exist with these games. Such issues can add frustration and slow progress, which overall makes the experience more challenging to engage with. P3 (*Foldit*, Expert), for example, comments on how they sometimes wait for hours for the game to process: "You know, when somebody says, oh, I just did this and it came out like that and then I try it and, you know, four hours later and I'm still not getting a result. It's very frustrating. And I have a good machine." P5 (*Foldit*, Novice) expands on this, noting the ways in which frustration with technology interacts with the previously described gameplay confusions: "And if you don't know what you're doing and you poke at something for half an hour and your computer's running really slow, you really haven't made very much progress."

These two feelings: confusion and frustration, multiply against each other as the technical barriers of the game interact with the design barriers. Compare this to Paavilainen's modern definition of playability as functionality, usability, and gameplay [375]. Our findings map directly to Paavilainen's model as we have divided the Missing Polish into usability (UI), gameplay, and technology (functionality).

The exploratory learning phase of gaining expertise involves interacting with the game artifact. Yet, interaction involves both the gulf of execution and the gulf of evaluation through the UI and the gameplay, both levels mediated by the technology of the software itself. Frictions with the UI, the gameplay, and the technology can each hinder exploratory learning. These frictions lead to breakdowns in action, understanding, and eventually involvement [228].

Each of these frictions can be addressed in turn: ECCSG developers can collaborate with professional UI/UX designers to create clearer user interfaces; professional software developers can help optimize performance and reduce bugs; and professional game designers can help develop more intuitive gameplay using industry principles of tutorial design, level design, and gamification design.

5.3.4 Missing Communication (The Sociocultural Barriers)

Lastly, given the social nature of learning for ECCSGs, expertise depends heavily on strong, open communication. However, we found ECCSG communities largely lacking this communication due to factors such as a lack of adequate community content, gatekeeping, jargon, silence from the developers and scientists, and competitive restraints. These communication barriers prevent social learning, the second half of the explore-discuss cycle of learning. We examine each of these sociocultural issues in turn, separating them by player-player communications and player-developer communications.

5.3.4.1 Player-Player Communications

First, there is not enough high-quality community content to scaffold learning. Other games, specifically popular commercial games, have in-depth wikis and dedicated content creators, such as YouTubers and Twitch streamers. Yet, this community content does not exist to the same extent for these CSGs, and furthermore there are gatekeepers and other barriers to enabling players to become content creators and address this inadequacy. Earlier, we described *Minecraft: Education Edition* as having more structured onboarding, but the commercial version of the game is still taught through a similar manner to ECCSGs (exploration and social learning).

> *Minecraft is a good example of a game where you basically learn the game from the wiki or you watch a YouTuber play it, you know, and you know, with Foldit, you don't really have that. You don't really have a lot of video content creators for the game.* (P1, *Foldit*, Intermediate)

With the amount of community content that exists for games like *Minecraft*, new players are easily assimilated into the affinity space of the game through a large number of "portals" (streamers, YouTubers, wiki guides, etc.) [179]. But, as P1 notes, ECCSGs have nearly no content creators. Furthermore, not only is the quantity lacking for ECCSGs, but the quality of the content itself is often poor due to a combination of low-quality recording hardware and editing software, inexperienced content creators, and complex game mechanics being discussed.

> *I'm done trying to find videos... I tried to watch something on YouTube. I tried to find some videos and I [found] a few tutorials. I watched them and they [were] in very, very ugly quality. And a few of them [were] too complicated for me, who just started ... playing.* (P7, *Foldit*, Intermediate)

In essence, ECCSGs are following a model of cognitive apprenticeship [91], wherein the professional vision is modeled by content creators and observed by new members to the affinity space. This same model has recently become popular on Twitch for domains such as coding [157] and eSports [182]. Similarly, wikis also provide a portal into the affinity space and allow learners to co-construct knowledge as another form of cognitive apprenticeship [578]. Yet, although ECCSGs are following this practice, they do not have the same level of organized, published information or archives of content that more successful domains have. P13 notes this when comparing *Eyewire* to the board game Go:

> *So when I'm learning both Go and Eyewire, I assemble heuristics and create my decision tree based on those rules... For Go, is, is an ancient game and there are quite a few books and, rules of thumb, heuristic type things that have been developed by a lot of people... For Eyewire, the heuristics aren't really listed anywhere.* (P13, *Eyewire*, Intermediate)

On the surface, it seems that both the low quality and quantity of community content could be attributed simply to a smaller community size. However, upon further investigation, this was not the only factor — ECCSGs have specific barriers further preventing content creation. When players were asked why they do not contribute to creating content themselves, several typical responses were given, such as a lack of time (e.g., not being able to fit it into their schedules). Yet, two other responses were frequently given which appear to be unique to expertise-centric games: a sense of inadequacy and a fear of how the content would be received.

> *I'm not really a guy that tries to go out and make videos real quick and edit wikis until I know that I'm very knowledgeable on what I'm talking about... Right now, I'm not comfortable with like showing how I'm playing... And I would not be comfortable trying to edit the wiki. These guys [other players] know way too much.* (P11, *Foldit*, Intermediate[5])

Previous literature has also identified learner confidence as a barrier to making contributions to community content [299]. Furthermore, a study of Wikipedia found several related factors important in making contributions, including (among others) a sense of belonging, altruism, attitude, subjective norms, and knowledge self-efficacy [79]. It seems that in the case of ECCSGs, knowledge self-efficacy (causing sense of inadequacy) and sense of belonging (causing fear of reception) are two potentially limiting factors inhibiting the intent to contribute. This is further evidenced by P13 (*Eyewire*, Intermediate), who said he didn't make content because he was "a little worried about stepping on toes," a sentiment echoed by another participant who described "Getting to the point where I knew enough of the players" (P16, *Eterna*, Intermediate) as a prerequisite to creating community content. To these players, having a strong social network and the social status associated with it — i.e., a sense of belonging — appears to be a requirement for creating content (as well as for extended participation, cf. [21]).

5.3.4.2 Player-Developer Communications

In addition to these two factors, knowledge self-efficacy and sense of belonging, players can also be pressured by explicit gatekeeping from developers if their contributions are not perceived as accepted or acceptable:

> *I've thought about reorganizing the Eterna wiki. But there's a lot on there and it's hard to find, not very well organized... It's a ton of work, and I have a feeling, from what I've*

[5] Recall that intermediate players typically have *years* of experience. The fact that they still do not feel able to create content of any kind is striking. This opinion was expressed by several intermediate players.

> *seen, it would be dismissed... Probably wouldn't be accepted... It probably would be wasted time... Probably no one else would agree on it and it would never go anywhere... I've seen a lot of other players spend a lot of time creating content and tutorials and organizing things, and it just gets ignored. [Interviewer: Does it get ignored by the players or the developers or both?] The developers.* (P16, *Eterna*, Intermediate)

We also observed that some of the participants who noted this gatekeeping were women. A recent study on Wikipedia contributions found a gender bias due to a "vicious circle" of negative reputation, anonymity, fear, alienation, and rejection [301]. The authors note that several sociocultural barriers occur both pre- and post-contribution. A similar set of issues may be occurring in ECCSG community content contributions as gatekeepers prevent women from contributing knowledge in these spaces. Recognizing player contributions is therefore critical for supporting community content, especially because it speaks directly to the motivations of players who want to be recognized for their contributions in the CSG [229, 114, 241, 136].

Our findings regarding social learning relate to the Creativity component of the MLC model [241]. As Jennett et al. write, online citizen science learning is "informal, unstructured and social," and it follows a virtuous circle: "a volunteer improves her knowledge and skills by doing the task, sharing this in a community of peers helps to increase her self-confidence, also increasing her ability to perform the task and her desire to share ... the community helps her to become more competent, which will finally enable her to help newcomers in the community, therefore becoming conscious of her learning and more self-confident in both performing the task and assuming new roles in the community" [241, p. 15]. Yet with ECCSGs, we found that a lack of knowledge self-efficacy and sense of belonging inhibit players from sharing back to the community, thus breaking this cycle and preventing community-based learning. In fact, it's possible (though left for future work) that the only causal factor here is knowledge self-efficacy — that a low sense of belonging is a symptom rather than cause of this barrier. Indeed, by definition, most of the differences between ECCSGs and other citizen science projects is the emphasis of expertise, which is what appears to be triggering the low knowledge self-efficacy and could hardly explain the low sense of belonging without other mediating factors.

Moreover, although it is clear that social learning is a critical factor, participants expressed dissatisfaction with the current channels of communication. This includes player-player communication, such as chat channels being too quiet (P4, *Foldit*, Expert). However, more often participants discussed issues with the communication between the players and the science/development team. For example, scientific jargon is a strongly demotivating factor for novice players:

> *I try to [follow Foldit news]. But then to a certain point, you know, [the news post] starts to talk about things which I just don't understand. And then, you know, it just kind of... at that point I don't pay much attention.* (P6, *Foldit*, Novice)

Novice players often don't have the prior knowledge to understand the jargon being used to describe scientific progress. Moreover, this struggle is not restricted to novices — even players with years of experience feel doubt about engaging with news updates from the scientists.

> *[On the subject of when communication breaks down] The short answer on this is: jargon. They're using scientific jargon that we don't understand... Players have to ask a lot of questions, because it's unclear to us. We have to ask a lot of questions. And I'm the, I end up being the one who asks a lot of questions. And I don't know if that's because it's just not occurring to other people to ask these questions or they're embarrassed and don't want to look stupid, but. I, and even I at some point hesitate to ask any more questions because, like, I'm worried, it's making me look uninformed... to be asking too many questions.* (P16, *Eterna*, Intermediate)

Studies on scientific communication have found that jargon can reduce perceived and actual understanding and can even affect scientific interest, information-seeking behavior, and potentially self-identification with the scientific community [475, 66]. Because of their deeply-ingrained expertise (and lack of public communication training [376]), scientists may struggle to avoid jargon while communicating [417] (although some evidence suggests that scientist communication can be a viable alternative when professional communication is not available [34]). However, this is assuming that the scientists communicate at all. In some cases, players are simply disappointed with the frequency and transparency of scientific updates:

> *I also am concerned that the science behind the... We never, we don't see hardly any results of this development... They occasionally will come out once or twice a year and say we mapped this neuron and show a picture or a collection of neurons. But nobody's really saying what actually they are learning from it or what they are trying to learn from it.* (P13, *Eyewire*, Intermediate)

A lack of communication, especially regarding the outcomes of player contributions, can be seriously demotivating for players. Yet, even before players attempt to engage with blog posts and other updates from the science team (where present), confusions already arise from the game itself. Players have outstanding confusions about the citizen science components of ECCSGs and how their play affects scientific research. The game-science research loop — the fundamental core of ECCSGs — is not described adequately to them:

> *I'm very iffy on what it means to create a solution in Foldit and then somehow that goes into the lab... I still don't really know much about how that happens.* (P11, *Foldit*, Intermediate)

Given that the premise of CSGs is that anyone can contribute to scientific progress, one would expect that the game itself would adequately explain everything the player needs to know about their contributions; the player shouldn't need to seek out other information sources on the contribution model, yet players are going years without understanding the game they are contributing to. Similarly, there is little to no feedback to the players about their performance, so they feel confused about whether they are making meaningful contributions as they play. "It's difficult to tell [that] what you're doing in the game matters at all..." says P8 (*Foldit*, Novice). Compare this to the earlier finding that making meaningful scientific contributions is a core motivating factor for sticking with an ECCSG. If players cannot understand how their contributions are meaningful, then they cannot value the game for making contributions, and so they are much more likely to abandon the game.

These results agree with the findings of Díaz et al. [136]. For example, one of their participants said their experience was "Fun but frustrating, it would be nice to have a better understanding of how the data helps real life research. Enjoyable game, but I lost interest due to the perceived disconnect from the science behind it."

Scientific communication is difficult, not just in CSGs but in citizen science more broadly [439, 376] (and, indeed, in all public communications). For in-depth solutions, we refer to Rüfenacht et al. [439] who recommend, among other steps, appointing a communications expert to the core team and developing a communication and dissemination strategy.

Although we have divided this analysis between player-player and player-developer communications, all communications are ultimately shaped by the design and development of the project. The player-player communication barriers — most notably low knowledge self-efficacy and low sense of belonging — are entangled with the instructional barriers and influenced by the project's onboarding design. Thus, despite developers not being directly involved in interplayer communication, they still have the power to moderate, mediate, and manipulate it.

5.3.4.3 Tension between open science and secret competition

Notably, within *Foldit* only, participants described a tension in the game that stems from the dynamic of groups (teams) which are both private and competing.

> *So people are still reluctant to share certain things outside of the group setting. So, you know, where, in group [chat], I might be talking about some overall strategy about low energy or clash importance, in veteran [a more public chat channel] I wouldn't talk about that at all. And, you know, I've had people in veteran say, well, how did you do that? And my first knee-jerk reaction is to tell them. And then my second reaction is, well, wait a minute. You know, that's something my group would want to know, but I'm not sure I want to tell you. And, you know, it's science so we all benefit. So why don't we all do it? So there's that tension between the two.* (P3, *Foldit*, Expert)

From a novice perspective, and indeed from the perspective of a collaboration toward open science, the idea of hiding knowledge for personal gain is paradoxical to the ECCSG's mission.

> *I'm a new player. I would benefit greatly from new solutions, but I don't have access because I might compete with this player and my group might benefit from theirs... There is some level of weird sort of friction that's like, I can't receive help from you because you're on the other group... There's such a level of secrecy placed on sharing solutions that it's completely unwarranted.* (P8, *Foldit*, Novice)

This tension was previously identified by Ponti et al. [392], so our purpose here is only to confirm their work, not to unveil a new phenomenon. Yet, we highlight it here because it may speak to a larger tension: the tension between *game* and *science* in citizen science games. Of all dilemmas (in the Triadic Game Design sense of the word [206]), the tensions between game and science seem to be most often in conflict for ECCSGs. We speculate this is due to the lack of specialized persons with both scientific and game design expertise, as suggested by Prestopnik and Crowston [400]. It is unlikely that these two domains are inherently incompatible; rather, CSG development teams have historically lacked sufficient expertise in one or the other. In any case, the takeaway message from this is to approach competition mechanics with caution, since competition is the driving force creating secrecy in *Foldit*.

In summary, we see several major barriers to social learning in ECCSGs: small communities, low knowledge self-efficacy, low sense of belonging, community content gatekeeping, inaccessible scientific communication, infrequent scientific communication, and the tensions of competition. The issue of small communities is confounded with other factors that make ECCSGs a niche community, while the issues of low knowledge self-efficacy, low sense of belonging, and community content gatekeeping (i.e., developer dismissal of paratext contributions) appear to be unique to ECCSGs and a novel finding of this work. These barriers may be lowered by making community engagement more accessible, such as by providing technical and social assistance for creating content and encouraging forms of engagement that don't require expert knowledge. The next two barriers, inaccessible and infrequent scientific communication, are known problems in the broader fields of citizen science and general science dissemination, and solvable using recommendations from prior literature, such as regular communication schedules [439, 376, 475, 417]. Lastly, the tensions between group competition and open science were first identified by Ponti et al. [392]. We expand on their finding by placing it in the context of a broader tension between the *game* and *science* of citizen science games, as also seen in Section 5.3.1.1 on the various ways science knowledge interplays with game expertise.

5.4 Discussion

Through interviews with ECCSG players of varying expertise, we generated four themes to describe the path to expertise and barriers in this path. The first barrier players encounter is Missing Instruction. Then, as players begin a cycle of exploration and social learning toward expertise, they encounter the barriers of Missing Polish and Missing Communication which inhibit their exploratory and social learning respectively.

The golden path to expertise is social and exploratory. One's initial engagement may be driven by an interest in contributing to science or for personal reasons (such as entertainment or personal learning), but continued engagement depends on both of these motivations being present simultaneously. Once expertise is reached, one gains a "professional vision," or an understanding of seeing, discussing, and thinking about the game that is shared with other experts [395, 189]. This reflects the case study of Apolyton University — a player-made online community of *Civilization* players [486]. This community's online discussions of incredibly nuanced topics and their protocols for novice player onboarding, as Squire describes, is an exceptional example of knowledge production and organization, design thinking, and social learning with cognitive apprenticeship. Squire's case study demonstrates that game communities are capable of managing extreme expertise. Here too, we see an exploratory, social approach to gaining expertise and a shared understanding of that expertise within the community.

As such, for CSG developers to teach in the ways that players learn, we recommend leaning into the social and exploratory aspects of game-based learning. This can include, for example, social features to better enable inter-player communication, in-game wikis, and modes of play that encourage and support exploration. Moreover, we encourage CSG developers to appoint professional community managers to maintain the social spaces of their community, as suggested by other citizen science scholars [439, 544, 282].

The first set of barriers on this path is instructional, introduced by a lack of clear goals and strategies. First and foremost, players in this study and in prior literature (e.g., [136, 395]) widely agree that ECCSGs have incredibly steep learning curves on initial entry. Paradoxically, the tutorials are simultaneously too simple and the gameplay too difficult, namely because the tutorials focus on micro tasks and basic control schemes without introducing more advanced, nuanced concepts that are required for standard expert play.

This design paradigm creates what we will call the "**giant's staircase**," where difficulty begins flat and trivial and suddenly spikes to insurmountable heights. With complex games like *Foldit*, there may be several such levels of giant's steps, meaning that even if you can overcome one giant step, there may be more remaining between intermediate and expert play, alternating between trivial and impossible.

The lack of teaching all critical concepts is especially noteworthy in that these games fail to teach the big picture of what the game is even about and how playing is contributing to science. By focusing on the micro tasks, it is not made clear to players what the macro tasks are and how these skills and tools connect to a broader impact. Lastly, there is a distinct lack of feedback given to players to help them improve their play. This is in part due to the qualitative and unknown nature of scientific contribution, yet there exist current and potential social and ludic dynamics which can overcome this challenge and provide feedback, as noted by player suggestions during the interviews.

Overall, our findings on the instructional barriers of ECCSGs agree with prior literature on the low efficacy of CSG tutorials and need for better explanations on the scientific contribution models [136, 477, 296]. We therefore recommend appointing professional instructional designers who can create effective training materials and professional game designers who can make such materials enjoyable. Moreover, given how important the big picture, core gameplay loop, and scientific contribution model are to understanding and contributing, it is critical to teach these early and refer to them often, such as via the "whole-task" approach of 4C/ID [540]. Lastly, for teaching advanced gameplay topics, collaboration with expert players is important to ensure that expert strategies are taught and taught well.

The second set of barriers on this path is game-based, introduced by a lack of polish on the game software itself. Confusions and frustrations are caused by a disorganized user interface, slow or aging technology, and unintuitive gameplay. Further, the gamification elements are sometimes in conflict with the scientific goals, leading players to behave ineffectively. The lack of clarity in goals may also unintentionally encourage players to play in a way that is not engaging for them, further defeating the purpose of the CSG.

As shown in prior literature, citizen science technologies often suffer from poorly designed user interfaces [477] and technical issues which can reduce participation and data collection [534, 174, 470]. Similarly, historical attempts at gamifying citizen science in ways that aren't aligned with scientific goals (e.g., competition) can lead to disinterest and discontinued participation [154]. Citizen science games are "system assemblages" of many technologies, which requires careful attention toward integrating these systems in order to support participation and the scientific goals of the project [406]. We therefore suggest employing professional UI/UX designers, software developers, and game designers to provide the necessary polish required for an engaging and operational CSG.

The third set of barriers on this path is sociocultural, introduced by a lack of open communication. Discourse is limited by incompatible sets of jargon between players and scientists, and restricted by low knowledge self-efficacy, low sense of belonging, and developer gatekeeping. Fear of gatekeeping and a sense of inadequate expertise and belonging prevents the creation of community content, further limiting knowledge sharing. Moreover, players expressed dissatisfaction with the amount of feedback they receive from scientists and developers. Within *Foldit* specifically, there is also a tension between the secretive, competitive gameplay and the overt goal of open science contributions, as previously noted by Ponti et al. [392].

These findings agree with recently discovered mechanisms of community content contributions [79] and scientific jargon [475, 66]. Therefore, we recommend CSG developers provide technical and social assistance for encouraging community content while discouraging and preventing gatekeeping. Working with community managers and science journalists can help the team create and execute a strong communication and dissemination strategy and become more accessible and transparent in their scientific communications, as detailed in recent citizen science publications [439, 376].

We also note the entanglement of these three sets of barriers. Although each barrier is a distinct cause of problems from a separate design space, together they overlap in symptoms and create interaction effects. For example, consider *Foldit*'s "finicky" tutorials. On the surface, this issue is caused by inconsistent game mechanics or otherwise poor game design. Yet, the issue would not be so frustrating if the game also provided clear and thorough feedback or access to high-quality community-made guides to help players reach the goal. Meanwhile, players are simultaneously struggling with a complex UI, slow technology, a steep learning curve, etc. In this way, removing only one of these barriers may not even resolve the issues, or may resolve one but surface another.

This work supports previous literature on player experiences in CSGs. As found by

Eveleigh et al. [153], we observed that intrinsic motivation leads to deep engagement, and that initial participation may be increased by acknowledging contribution efforts and enabling the players to fit the game around their existing schedules. Like Díaz et al. [136], we found that players thought the tutorials were too simple and the game too difficult, and that players want more explanation of the science and the overall goals of the project. Finally, echoing Skarlatidou et al. [477] and Spiers et al. [484], players expressed desires for better tutorials and larger, more diverse communities.

Prior research shows that game communities use games to learn experientially via active and discovery learning, forming affinity groups, cycles of expertise, well-ordered problems, and simplifying conditions [175, 178, 180, 426, 486]. In ECCSGs, we respectively saw exploratory learning, community knowledge building, cycles of exploration and social learning, and tutorials which attempt to simplify and order the problem space (despite their failings described in Section 5.3.2).

Similarly, our findings are consistent with Jennett et al.'s MLC model and their barriers to engagement, which included difficult or boring tasks and lack of time [241]. We found lack of time cited as a reason for not contributing community content and difficult tasks to be the primary symptom of instructional failings. These results are also in agreement with Aristeidou et al.'s factors of participation, namely lack of time, website usability, fear (of engagement), quality of contributions, and sense of belonging [21]. Specifically, we found that low knowledge self-efficacy was a unique factor to ECCSGs and likely drove fear of engagement and sense of belonging.

The rest of this section places this work in broader contexts and discusses implications. First, in the broader context of CSGs, to what extent does expertise matter at all? As suggested by Eveleigh et al. [153], perhaps some participants don't want to be experts, only dabblers. However, dabbling appears to be incongruous with ECCSGs. Unlike data-centric CSGs where contributions are proportional to effort, ECCSGs have an exponential curve with respect to effort versus output: a great deal of investment at the beginning of one's play will amount to little usable output, but after initial investments to learn the domain, even a small amount of effort will produce valuable contributions. In this way, as ECCSGs are currently designed, the effort of dabblers (excluding those with pre-existing expertise) is better spent elsewhere, such as on data-centric projects.

	Findings and Recommendations	**Expected Generality**	**Contribution**

Path to Expertise	**Learning is exploratory and social**	Broad	Vali-dates [241]
	Learning is a *cycle*: exploratory *then* social Improve social learning by adding or improving social features for connecting players both in-game and through other channels, such as wikis, forums, Discord servers, and community content. Collaborate with professional community managers. Improve exploratory learning by adding ludic features which encourage exploration and intelligent trial-and-error.	ECCSGs	Novel, cf. cycles of expertise [178] and GIIL [230]
	Motivation requires both meaningful contribution and self-gain Consider how the game feeds player motivations for contributions to science, entertainment, and personal learning; provide features to address all of these motivators.	CSGs	Extends [114, 241, 513]
	ECCSG players develop a "professional vision"	ECCSGs	Extends [395]
Missing Instruction	**Entry skill barrier** Collaborate with professional instructional designers and game designers to provide effective and enjoyable task progressions.	ECCSGs	Novel
	Lack of feedback Provide social features for in-depth peer-to-peer feedback. Provide dynamic automated feedback within the game. Provide frequent communication from the scientists regarding how contributions are being used and the results of scientific analysis.	ECCSGs	Novel, cf. [151]

	Part-task approach Teach the big picture, macro tasks, core gameplay loop, and contribution framework early.	ECCSGs	Novel, cf. [540, 151]
	Ends instruction early Collaborate with expert players to understand and teach the advanced techniques required for intermediate to expert play.	ECCSGs	Novel
Missing Polish	**Unintuitive UI/UX** Collaborate with professional UI/UX designers to develop clear, organized user interfaces.	CSGs	Extends [136, 374]
	Technical issues Collaborate with professional software developers to optimize performance for typical usage, especially novice play.	CSGs	Extends [136, 374]
	Unclear gameplay Collaborate with professional game designers to develop clear and intuitive gameplay using industry principles of tutorial and level design.	CSGs	Extends [136, 374]

Missing Communication	**Low knowledge self-efficacy** Make contributing community content more accessible for all players by providing technical and social assistance.	ECCSGs	Novel, cf. [204]
	Low sense of belonging Detach perceived game skill from perceived ability to contribute by encouraging other forms of community content contribution, such as fan art and social events.	ECCSGs	Novel, cf. [36, 37]
	Community content gatekeeping Explicitly address, discourage, and prevent gatekeeping.	ECCSGs	Novel, cf. [301]
	Insufficient scientific communication Provide more details on the scientific topic and acknowledgements/praise of player contributions with details of how their efforts translate to scientific advance. Invite open communication. Collaborate with communicators and science journalists to provide clear, public-facing translations of scientific jargon.	Broad	Validates [439, 376, 475, 417]
	Tensions of competition Design for collaborative rather than competitive gameplay. Look for and eliminate game dynamics which encourage secrecy.	*Foldit*	Vali-dates [392]

Table 5.1: A summary of findings and recommendations addressing the issues identified in the reflexive thematic analysis.

Next, there is the "elephant in the room" issue noticeably brought up by our findings — if the solution to many of the barriers to expertise is to hire professionals with varying skill sets, how ought CSG teams accomplish this with the current state of funding? Typically, CSGs are funded much like educational technologies via research labs and government grants [270]. To a lesser extent, they can also be funded by private donations; and although some citizen science projects are funded by entrepreneurial sources such as participant fees and merchandise sales, no CSGs (to our knowledge) employ that participant-funded model at this time [560]. CSGs therefore have no sustainable financial model, especially given that research grants rarely cover long-term project maintenance [270]. Klopfer et al. [270] posit that educational technologies might look to entrepreneurial incubators and startups to help turn their products revenue-positive in order to become sustainable, perhaps even employing a hybrid model where technologies are prototyped via research grants and polished via entrepreneurial partners. However, transferring this model to CSGs would raise new ethical questions if, for example, the ability to contribute to scientific knowledge becomes locked behind paywalls. In short, there is no easy solution to broadening CSG teams to include a wider array of expertise given the current financial infrastructure. Instead, we suggest CSG teams begin investigating these alternative funding sources (exploring their ethical implications beforehand) and secondly reach out to experts who may be willing to offer some *pro bono* assistance.

Our study answers an important research gap: why is ECCSG onboarding currently insufficient, and in what ways? Specifically, this work provides two major contributions to this question. First, we present a model of how expertise is acquired in ECCSGs and the barriers along that path. This extends previous work on CSGs with respect to understanding expertise [255], player experiences [136], and skill acquisition [334] and adds a specific framework for how skill-based expertise is acquired and the ways in which its acquisition is hindered. Second, we identify three barriers to expertise and unpack detailed mechanisms of how they interfere with learning and engagement. With respect to instruction, the mechanisms are: high skill requirements on entry, lack of feedback, lack of a big picture explanation, and lack of intermediate-to-expert instruction. With respect to the game artifact, the mechanisms are: unintuitive and cluttered user interfaces, software issues, and unclear gameplay. With respect to the interpersonal, the mechanisms are: low knowledge self-efficacy, low sense of belonging, content gatekeeping, inaccessible and infrequent scientific communication, and game-science tensions.

Notice also that — when taken abstractly — only the sociocultural barriers are specific to the unique science-game model of ECCSGs. The game-based barriers (poor UI, software, and gameplay) could be found in any game, and the instructional barriers (poor instructional design) could be found in learning any topic for which learning materials do not exist in abundance. Therefore, we expect this work to be useful as a guide to product refinement for other fields of game-based learning, such as the iterative development of educational technologies and serious games.

We provide a summary of our findings in Table 5.1, noting where our contributions are novel or extend/validate prior literature. Importantly, because this work is strongly interdisciplinary, many of our findings can be linked to prior theory in learning sciences. While we aimed to note as many connections as possible, what we list as novel contributions is meant to highlight original observations specifically in ECCSGs.

5.4.1 Limitations, Implications, and Future Work

This study is not without limitations. The first and foremost limitation is a small and mostly homogeneous sample. More evidence is needed to confirm this model, especially in *Eyewire* and *Eterna*. Moreover, despite its potential, there is little evidence to support generality outside of ECCSGs, which are quite a niche field. Nevertheless, this exploratory work provides direction for future research in dismantling the barriers and accessibility issues which prevent citizens from contributing human computation and creativity toward scientific advancement. As the 21st century continues to show a need for effective mass cooperation and interpersonal efforts in a variety of ways, this work contributes toward our understanding of interdisciplinary and complementary expertise in the example of game players gaining and contributing expertise to scientific research.

Though this work is meant for ECCSGs, other game-based learning projects may benefit from understanding the barriers to player learning. Projects which meet any of the expertise-centric criteria — from gamified language learning to game-based algorithm discovery — can learn from the ECCSG model how to better empower their users' learning and the ways in which learning and contributing are suppressed.

What are the takeaways and implications for the future of ECCSGs? Considering the arguments of Keep, it is critical to focus on improvements to knowledge sharing, knowledge organization, and discourse translation between players and scientists [255]. Moreover, as discussed in Section 2.3, this work suggests possible ideas in the design space of ECCSGs. Any projects needing the affordances of ECCSG criteria — addressing a system-driven, self-contained, complex problem space with many instances of problems to solve — may consider making their project into an ECCSG.

For future work, the next step for improving ECCSGs is to remove these barriers via design-centered research. A table of recommendations for topics of improvement based on these themes is provided for CSG developers in Table 5.1.

5.5 Conclusion

ECCSGs can be a powerful tool for crowdsourcing scientific advancement, yet in practice they are incredibly difficult to design well, such that its players are trained to become experts. In this study, we interviewed ECCSG players of *Foldit*, *Eterna*, and *Eyewire*, then applied reflexive thematic analysis to generate themes of their experiences. The analysis produced a model of expertise in ECCSGs and barriers therein. The path to expertise was found to be a cycle of exploratory learning — yet hindered by a lack of game polish — and social learning, hindered by a lack of communication. Entrance to, and repetition of, this cycle is barred by poor instructional design, nicknamed the "giant's staircase." This work validates and extends several previous studies on player experiences in CSGs. Based on this work, we call into question the current financial and participatory models of CSGs and make recommendations for CSG developers, including collaborating with professionals of required skill sets, providing social features and feedback systems, and improving scientific communication.

5.5.1 Takeaways

This study gave us a clear map for how players learn to become experts in ECCSGs: they explore, they socialize, and they iterate. We identified some specific issues that prevent their learning, and for the most part they align with what we found in Chapter 4 as well: players need a well-polished game, clear and helpful instructions, and strong communication on what they should be doing and how their efforts are contributing to scientific research. With these insights laid out, I have — for the most part — achieved the goal of identifying problems with onboarding design in expertise-centric citizen science games. However, before we call this a mission success, we need to validate that these issues are truly what's creating problems for ECCSGs. Furthermore, we should test whether we can fix any of these problems now that we have identified them.

This brings us to Part III. What are the solutions to the problems in onboarding design for ECCSGs? Before we can directly try to implement something, we'll need to gather data first: what do good solutions even look like? Let us make sure that whatever we implement is grounded in practical methods. So, in the next chapter, I analyze the onboarding design of three "camps" of games — citizen science games, educational games, and commercial entertainment games — to understand the trends of tutorial design and the successful (and unsuccessful) patterns of game-based learning.

Part III

Finding Solutions

Chapter 6

Tutorial Design Patterns

Now that we understand the problems of onboarding design in ECCSGs (RQ1), the next step is identifying solutions (RQ2). Toward this goal, I and my co-authors investigated what makes CSGs different from other games by comparing the onboarding design of CSGs to that of educational games and commercial entertainment games. The idea in this methodology is that we can learn from other development "camps" or design ideologies to look for successful strategies that can be applied to CSGs. This study is also useful more generally for improving our understanding of onboarding design.

Typically, onboarding design research focuses on specific elements instead of the holistic experience (e.g., context sensitivity in VR [169], three-star systems [173]) or on a specific game instead of larger populations (e.g., *Saber's Edge* [344], *BeadLoom Game* [471]). Moreover, little research has been done to check if commercial entertainment games onboard their players differently than in serious gaming. Successful commercial entertainment games are (tautologically) successful at onboarding players — and because serious games need their players onboarded to achieve the serious purpose of the game — it would benefit the serious gaming community to look at what makes entertainment game onboarding successful.

Therefore, in this study, we compare educational games, citizen science games, and commercial entertainment games as representative of three distinct camps of design. While there are others, we focus on these as representative of communities of practice [180] which have strong reasons to be concerned with learnability, approachability, and onboarding design.

I have already extensively described CSGs as a branch of serious gaming focused on scientific outcomes [97]. For these games to achieve their goals they require a large, trained audience — requiring both learnability and approachability — yet, due to their scientific topics, they can

‡Parts of this chapter were adapted from an article currently under review.

often be difficult to teach [337, 332, 401]. In this study, CSGs represent games which attempt both to teach complex subjects and to reach wide audiences.

Educational games capture the larger trend of serious gaming and game-based learning (GBL) (cf. [270, 271]). These games are worthy of study because they are explicitly focused on learning — often required by students for a class — and thus have the most reason to design for strong learnability.

Contrast both of these camps with commercial entertainment games, which have the resources, expertise, and incentives to produce high quality games that reach a wide audience for sales. Thus, this camp should demonstrate polished and approachable onboarding.

To make this comparison, we closely examine a variety of games in each camp using a close reading (or close play) method adapted from literary analysis [44]. This method allows us to deeply engage with the games to understand semantic and latent features which contribute to the onboarding design of the games.

We summarize our findings through three themes: (1) *Successful Games Pace Learning and Check Understanding*; (2) *Successful Games Set Expectations*; and (3) *Polish Means Playable Not Pretty*. We then compare the 39 codes used to generate these themes to other onboarding design literature (e.g., [130, 396, 512]).

This work provides two major contributions: first, our codes and themes validate and extend prior onboarding design research by breaking down heuristics into more detailed patterns. Second, by comparing patterns across development camps, we provide targeted insights for CSG and educational game developers on what makes commercial entertainment games successful in onboarding players. We emphasize the value in using competence gates, gradually increasing complexity, simplifying mechanics, setting clear expectations, and fixing usability and playability issues. More specifically for this dissertation, we identify several areas of improvement for CSGs onboarding and recognize specific mechanics that can help, such as competence gates.

6.1 Background

6.1.1 Onboarding Design

Research on onboarding design was briefly summarized in Chapter 2, but here we describe some additional research relevant to this study.

Most research specific to game tutorials has focused primarily on effective learning, especially for educational games. According to this body of literature, tutorials should provide "immediate positive cognitive feedback that combines corrective and affective support, short bursts of just-in-time instruction with minimal text and prominent visual cues, and step-by-step scaffolds that fade into free play over the course of the exercise" [471, p. 7]. Other studies have verified these claims about the importance of cognitive feedback [55] and context-sensitive (just-in-time) support [169] — especially through tooltips [14].

A few studies have instead focused on heuristics for designing and assessing the game's approachability. This includes Desurvire and Wiberg's Game Approachability Principles (GAP) [128, 130], Thomsen et al.'s onboarding heuristics for free-to-play mobile games [512], and Poretski and Tang's design strategies for video game learnability [396]. We discuss these further when comparing our results to theirs in Section 6.4.

Significantly less academic work has been done to understand tutorials in commercial entertainment games; however, there are many reports from the games industry on successful tutorial design, which I referenced in Chapter 2.

6.1.2 Close Reading Methodology

Aarseth has proposed three methods of studying games [2]. First, one can study the artifact itself: understanding the rules and mechanics and talking to the developers about it. Second, one can observe others play and read their reviews or reports of their experiences. Third, one can play the game themselves. This last method, Aarseth argues, is most important because without playing the game, researchers are liable to severely misunderstand it.

Following Aarseth's recommendation, this study embraces play as a fundamental part of game analysis, combining it with a critical eye toward the design of the game itself. This approach is captured by the close reading method, adapted for game studies from literary theory [44]. A close reading is "a detailed examination, deconstruction, and analysis of a media text [...] close reading is a way of laying bare the faults and inconsistencies [... and] excavat[ing] previously hidden qualities of a media artifact" [44, p. 289]. This highly interpretive process requires an oscillation between experiencing the artifact as a naive player and as a distanced, objective, and critical researcher.

We follow the methods as described by Bizzocchi and Tanenbaum to close read (or close play) a set of games. Though not always named as such, this method has been used successfully in other studies [280, 495, 69]. Other similar methods include gameplay review, which takes the second of Aarseth's methods [48, 266, 267], and instructional ethology [38, 39], which aims to reverse engineer a game's design to understand its teaching methods.

6.2 Methods

6.2.1 Game Selection

The games chosen for study were selected using relevant metadata to narrow the search, as suggested by Tyack et al. [528]. We included only games that met these criteria: available in English, publicly available online for operating systems available to the researchers (Windows, Linux, Android, and iOS), is single-player or has a single-player mode, is a game (as opposed to gamified tasks, training software, or non-game simulation software), and takes at least 5 minutes for a close reading. We further excluded time or location sensitive games (e.g., location-based games) and games that require additional hardware (e.g., VR headset, a guitar). Selection occurred around June 2020 independently by each researcher.

We first selected citizen science games, since there are fewer of them than other types of games. For citizen science games, we required that they have a citizen science component and that they are available at `https://citizensciencegames.com/`, a popular site for citizen science which collects links to active citizen science games. Following these criteria, I and another researcher (whose role, due to logistical constraints, was swapped with my co-author Kutub Gandhi following the selection phase) selected citizen science games.

We then selected approximately an equal number of educational and commercial entertainment games relative to the number of CSGs selected. For educational games, our criteria included that the game is finished (not in a beta or "Early Access" release) and has no citizen science component. We selected games from Steam (`https://store.steampowered.com/`) using the "Top Rated," "What's Popular," and "Top Selling" lists for the "Education" tag. We aimed to match the genres of the citizen science games selected in approximate proportions by searching using tag combinations (e.g., "Education + Arcade"). We further aimed to avoid duplicate selections, such as by selecting two games of the same series. Lastly, we aimed to ensure that all educational games had both strong educational and strong ludic components.

For entertainment games, we ensured that the games had neither citizen science nor educational components and were finished (not "Early Access"). We wanted to avoid games with significantly larger budgets and development teams than the average citizen science game. As such, we limited our selection to indie games, since most to all educational and citizen science games would be considered indie in scope of their production. Similar to educational games, we aimed to select genres that proportionally matched our citizen science game selections, using Steam's "Top Rated" lists for "Indie" and specific tag combinations as needed.

We note that this selection process resulted in significant overlap in the independent selections for citizen science games, moderate overlap for educational games, and little overlap for entertainment games. This reflects the population sizes of each camp of games. Moreover, by having partial overlap in the games analyzed by the two researchers, we were able to sample a wider array of games while still having some shared experiences for comparison. After initial selection, some games were later discovered to violate our criteria or be unplayable due to major bugs; we removed these games from the final report. See Table 6.1 for a list of the games selected.

6.2.2 Protocol

Kutub and I played through the games selected in a randomized order (in the case of Kutub, his games were selected for him by a previous researcher, as noted above). Each game was played until the initial onboarding felt complete, defined by the gameplay feeling stable (having a "rhythm") and the core gameplay loop being understood. Playthroughs were limited between 5 and 120 minutes and recorded for later analysis. To assist the close reading, we used the experiential questions in Appendix E as prompts. In practice, these questions were reflected on approximately every 5-15 minutes where appropriate. In total, 55 unique games remained in the final dataset. They were played for 47.29 hours of recording (M = 0.62; SD = 0.27).

The results were then analyzed in five phases. The analysis was primarily deductive, semantic, and constructionist, and grounded in existing frameworks of tutorial design (see Section 6.1.1). First, I (the primary coder) transcribed salient (i.e., meaningful to our understanding of onboarding design) utterances and experiences from all recordings. Second, I condensed the data into more general semantic and latent codes which captured recurring sentiments, such as "narrative video intro," "on-demand help," "minor UX frictions," and "trial and error solving." Third, I generated a codebook for the most salient codes that captured the recorded experiences and tabulated the data according to the codebook. Fourth, Kutub and I both reviewed the table and agreed on appropriate codes for each entry, discussing as needed. This analysis resulted in 39 design features and 4 metadata features: genre, sub-genre, release date, and platforms. Lastly, we reviewed the dataset in aggregate to generate themes which represent our findings. The codebook, game metadata, and

	First Author	Both	Second Author
Citizen Science	*Quantum Moves 2* [464], *meQuanics** [199]	*Foldit* [531], *Phylo* [322], *Eterna* [487], *Eyewire* [408], *Forgotten Island* [81], *Decodoku** [355], *MalariaSpot Bubbles** [390], *Apetopia* [530], *Colony B* [323], *Skill Lab: Science Detective* [463], *Cancer Crusade* [340], *Mozak* [532], *EcoBuilder* [357]	*MalariaSpot** [389], *Quantum Minds* [462]
Education	*PC Building Simulator* [88], *Learn Japanese To Survive! Kanji Combat* [480], *TIS-100* [575], *Influent* [433], *Air Forte* [47], *President for a Day — Floodings* [469]	*Niche — a genetics survival game* [492], *Zoombinis* [506], *Sokobond* [8], *while True: learn()* [307], *Tyto Ecology* [232], *Odyssey — The Story of Science* [508], *Kerbal Space Program* [485]	*Project Hospital* [373], *InMind VR** [306], *Poly Bridge* [143], *Breaking Good* [468], *War Solution — Casual Math Game* [571]
Commercial	*Papers, Please* [1], *Game Dev Tycoon* [193], *Unheard — Voices of Crime* [359], *Opus Magnum* [577], *Zup! 2* [412], *The Room Three* [162], *LYNE* [510], *Gunpoint* [498], *Boson X* [231], *Ikaruga* [522], *Super Hexagon* [507]	*Mini Metro* [138], *7 Billion Humans* [518], *Lightmatter* [524], *Baba Is You* [212], *Plague Inc: Evolved* [356]	*Lazy Galaxy* [90], *Sea Bubble** [278], *Orbt XL* [5], *TowerFall Ascension* [155], *Ding Dong XL* [6], *Luxor Evolved* [348], *Don't Escape — 4 Days to Survive* [465], *Veritas* [184], *ChromaGun* [387], *Starbound* [80], *FTL: Faster Than Light* [493]

Table 6.1: Games selected. Games with an asterisk were removed from the final report due to being unplayable, inaccessible, or violating our inclusion/exclusion criteria.

tabulated qualitative analysis are available at `https://osf.io/ut4mg/`.

6.3 Results

In the coding phase, we generated 39 codes (representing positive, negative, and neutrally descriptive characteristics) conceptualized into 7 categories (see Appendix D for brief descriptions of each code):

- **Explanation**: Unexplained UI, Details Unexplained, Strategies Taught, Unclear Scoring, Forced Exploration, Citizen Science Explanation
- **Design**: Level / Environment Design, Puzzle Design, Mobile Design, Clear Design Language
- **Attention**: Signaling, Gestures
- **Scaffolding**: Dynamic Help, Hints, On-Demand Info, One Pager, NPC Mentor(s), Tooltips, Competence Gates, Good Feedback, Performance Benchmarks
- **Pacing**: Just-In-Time (JIT) Tutorial, Task Variety, Gradual Complexity, Camera Controls Interactions Mechanics, Mechanic After Mechanic, Systems Exploration
- **Polish**: Narrative Introduction, Aesthetic Polish, Customizable Character(s), Standards or Conventions, Can't Go Back in Tutorial, Technical Bugs, Assume Game Literacy, UI/UX Issues
- **Meta**: Separation of Tutorial and Game, Educational Value, Felt Competent After Play, Motivated to Continue Play

After coding the data, we generated three themes which highlight our findings: (1) *Successful Games Pace Learning and Check Understanding*; (2) *Successful Games Set Expectations*; and (3) *Polish Means Playable Not Pretty*. For a brief description of each game's close play — including the games' genres, basic gameplay mechanics, notable features, and our experiences with them — refer to Appendix F.

6.3.1 Successful Games Pace Learning and Check Understanding

First, a game's pacing makes a significant difference in how its onboarding is perceived. This is a known heuristic [275, 127] and is grounded in the psychology and neuroscience of managing the player's cognitive load [500, 215]. However, we saw a few patterns that separated the camps in this respect.

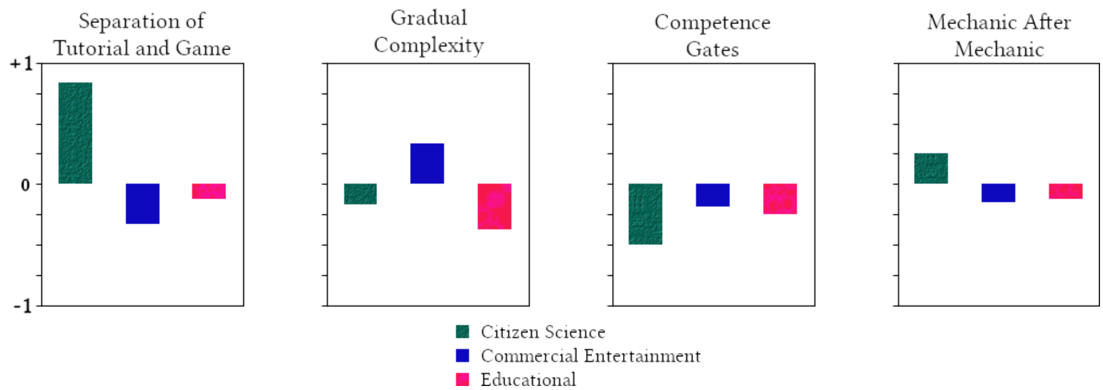


Figure 6.1: Game camps versus pacing features. The Y-axis represents the average code for each camp, such that a positive instance of the feature is 1, a negative instance is -1, and a neutral or uncoded instance is 0; each bar is the mean of all games in that camp. CSGs demonstrated significantly more separation of explicit tutorials and the actual gameplay or scientific challenges. Commercial entertainment games had a gradual increase in complexity more often than other games, especially compared to educational games. CSGs least often had effective competence gates, and more often introduced mechanic after mechanic without the opportunity for practice.

CSGs tried to teach many, many mechanics in their games. From quantum physics to protein design to image tracing, CSGs provided a plethora of tools and associated concepts needed to be competent at the game. And unlike the other camps, which intermixed tutorial and gameplay, CSG tutorials were often separated from the scientific aspects of the game. This resulted in a series of tutorial levels that would introduce mechanic after mechanic without practice (see Figure 6.1). This pattern was not limited to CSGs, for example these issues were noticed in *FTL: Faster Than Light* [493]. *FTL*, however, takes several measures to mitigate this: the ability and encouragement to pause, informative tooltips on all UI elements, reliance on genre conventions, and a gradual increase in complexity (the tutorial limits the type of ship and ship systems available to you — further play introduces new gameplay systems as you unlock content).

Entertainment games often took a gradual pacing to introducing new material, especially when compared to CSGs and educational games. Across genres, games like *7 Billion Humans* [518], *Mini Metro* [138], and *Starbound* [80] would give the player the smallest tool-set needed for core play before letting the player into the core loop of gameplay. Once the player is practiced and comfortable with the core gameplay loop, these games would introduce new elements one at a time and then hold back to let the player get comfortable with this new element before introducing something else. This is a known design pattern in the games industry [473, 497]. In fact, Nicolae Berbece summarized his 2016 GDC talk as “teach gradually through experience” [40].

Next, our experiences were often negative when there was a breakdown of understanding — a misunderstanding or confusion that halts game involvement [227, 228]. It makes sense, then, that games felt successful in their onboarding when they could identify or anticipate these breakdowns and take action to prevent or alleviate them. We saw instances of dynamic help, hint systems, and tooltips that served as scaffolding.

However, what assisted us the most were well-designed competence gates — challenges which prevent the player from advancing until they demonstrate an understanding of a specific mechanic or concept [563]. Competence gates were successful more often in entertainment games than CSGs. The most common failure mode of competence gates was that we were able to pass the gate with a misunderstanding or total lack of understanding. This led to a perceived sudden jump in difficulty when our flawed mental model would no longer permit us to bumble our way through. Another common failure of competence gates was tedium. In *Zoombinis* [506], for example, even when we understood how to solve a puzzle, executing the solution took a long time because of the slow animations. In *Opus Magnum* [577], on the other hand, the solution animation would stay slow just long enough for the player to get a sense of whether their solution was working (and, if not, why not) before gradually speeding up for convenience.

Giving the player feedback was a key part of checking for understanding. We observed several design patterns of feedback in this regard. Open-ended puzzles commonly had worldwide leaderboards, ranking systems, or optional challenges that players could use to benchmark their own performance against what was expected of them. In *Papers, Please* [1], a mistake results in an immediate citation describing what the player did wrong (and, thankfully, the first mistake is only a warning without penalty).

Feedback can also be provided through a scoring system; however, we observed that CSGs can have unclear scoring systems which create more confusion than feedback. In fact, 50% of the CSGs in the final dataset (n=12) had this issue. These games struggle because their scoring is grounded in a simulation of real-life mechanics which are not always transparent. For example, *Phylo*'s [322] scoring system is a modified scientific algorithm for calculating gene sequence similarity [254] — it aims to be scientifically accurate rather than easily interpretable; when we tried to align sequences we were confused why the score changed the way it did, since we had no understanding of the underlying mechanisms that contributed to the overall score change.

6.3.2 Successful Games Set Expectations

Because of our focus on a first playthrough, we purposely avoided learning too much about each game before playing. However, for some games, even after playing and looking up materials about the game, we had questions about the game's intentions. Most CSGs provided little detail about their exact research. For example, it is clear that *Apetopia* [530] is testing something regarding color perception, but why? For what research is this useful? We coded each CSG as providing specific, general, or minimal information about their research aims. Half of all CSGs (n=12) provided only minimal information, while the other half provided only general information; no games gave clear, specific details about how the gameplay data is being used for research.

Educational games, on the other hand, often had unclear expectations about the scope of their education. Although *Sokobond* [8] was tagged and themed as scientifically educational, it provided only forgettable trivia facts. *7 Billion Humans* [518], on the other hand, was not advertised as educational and yet covered a vast array of topics in program design. In this way, our analysis of the Educational camp was muddied by educational elements being present in CSGs and entertainment games while sometimes being trivialized in educational games. This trend also led to disappointment in mismatched expectations (see Figure 6.2). *Influent* [433], for example, advertises itself as a language learning game, when in fact it is only designed for the practice of listening and reading vocabulary words. Contrast this with *Learn Japanese to Survive! Kanji Combat* [480], which makes very clear from the beginning that its scope is limited only to learning Japanese kanji characters, and for this the experience was more coherent and enjoyable.

A similar issue affects both CSGs and educational games regarding simulations of real life effects (e.g., *Niche*'s [492] simulation of genetics or *Quantum Mind*'s [462] simulation of quantum physics). Although these games make some abstractions to gamify their subject, they are often non-explicit about how much is being abstracted: in what ways is the game like and unlike the real world? As a successful example, *while True: learn()* [307] has occasional pop-up screens which directly point out relations between game mechanics and real world analogs, highlighting their similarities and differences. Debriefing such as this makes a critical difference for the transfer of learning between the game and its real-life counterpart [106, 286].

Another pattern of unclear expectations regarded strategic decision-making, or *perceived* strategic decision-making. We noticed four types of interaction in tutorials which could be represented as a two-by-two matrix:

- **Decision + Strategy** — Skill-based play that the player should be learning how to execute in

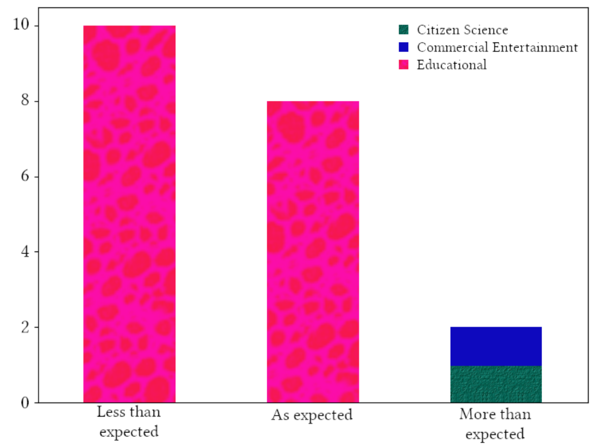


Figure 6.2: Educational value. This chart shows a count of games with educational content, coded as having more, less, or as much educational value as expected. A couple of games which were not listed as educational provided some educational content, while more than half of the educational games played did not provide as much value as expected.

order to get better at the game.

- **Decision + No Strategy** — The game offers a choice, but the choice is based on personal preference and does not affect the outcome of the game.
- **No Decision + Strategy** — Skill-based play that the game hand-holds the player through with the goal of learning by observation.
- **No Decision + No Strategy** — Operational play (core interactions) that the player should be learning to play the game at all.

Issues arose when the game did not properly communicate which of the four types of interaction was being taught. When we encountered a communication issue in this regard, our primary concern was "should I be spending cognitive resources here, and if so, what should I be putting resources toward?" Although the four interaction types described above may seem discrete, in practice there were several instances where what was being asked of the player was not obvious — leading to a frustrating misappropriation of cognitive resources.

These moments were often related to the codes "details unexplained" and/or a lack of "good feedback." An example of "details unexplained" was *Niche* [492], where a variety of genetic mutation options were presented to the player without clarification on whether the mutations were primarily cosmetic or had gameplay implications (and if so, to what extent). An example of a lack of "good feedback" was *Lazy Galaxy* [90], where a variety of machines were placeable by the player, but it was unclear how each machine was contributing to the score (relatedly, "unclear scoring" was also an issue in this regard). In both cases, the researcher (Kutub) was ostensibly succeeding at the gameplay, but was not confident enough in their skills to move on. They reasoned that, if they incorrectly moved on, they may end up confused later in the game. If they incorrectly stopped to examine each system in detail, however, they could waste cognitive effort on something ultimately cosmetic or irrelevant. In both examples, the lack of clarification led to frustration and a negative view of the game.

Improper framing of tutorial tasks can create extraneous cognitive load [500] during onboarding — extra load that can confuse players, lead them into misunderstandings, or cause frustration that cognitive energy was misappropriated. Tutorial tasks are designed with various intents: they can be a competence gate (testing basic understanding), a challenge (testing performance or the synthesis of mechanics), a puzzle (presenting an isolated problem with no future implications), or a lesson (presenting a problem whose solution will help with future problems). We describe these not to lay out a comprehensive framework but to show that each of these tasks requires a different mental approach from the player.

6.3.3 Polish Means Playable Not Pretty

Our third theme validates and extends my previous work (Chapter 5) which suggested that CSGs need polish. Here, we refine my earlier statement to specify that the polish which is missing in CSGs is a matter of usability rather than "game feel" [502]. We found that CSGs had more technical bugs and UI/UX issues than other games (see Figure 6.3). CSGs were also less likely to have narrative introductions and, curiously, far less likely to be available on desktop platforms and consoles. *Foldit* [531] was the only CSG with a downloadable desktop client, all other CSGs were either mobile and/or available in a web browser. Likely, this platform decision was made in order to reach a larger audience, yet because of the numerous technical and UX issues, players without the patience for bugs may churn out of frustration.

In our experience, a lack of aesthetic polish was a forgivable offense, but deeper questions on interaction led to frustration. Some games had repetitive sound effects, a lack of music, poor UI scaling or improper resolution, or default UI assets.

Yet, these issues were minor compared to the cognitive experience of onboarding and the core gameplay loop. If the game couldn't clearly answer the what, why, and how of interacting with the game, its aesthetic polish was irrelevant: the game was already frustrating. And on the other hand, if gameplay interactions were smooth and enjoyable, we would likely have been able to accept

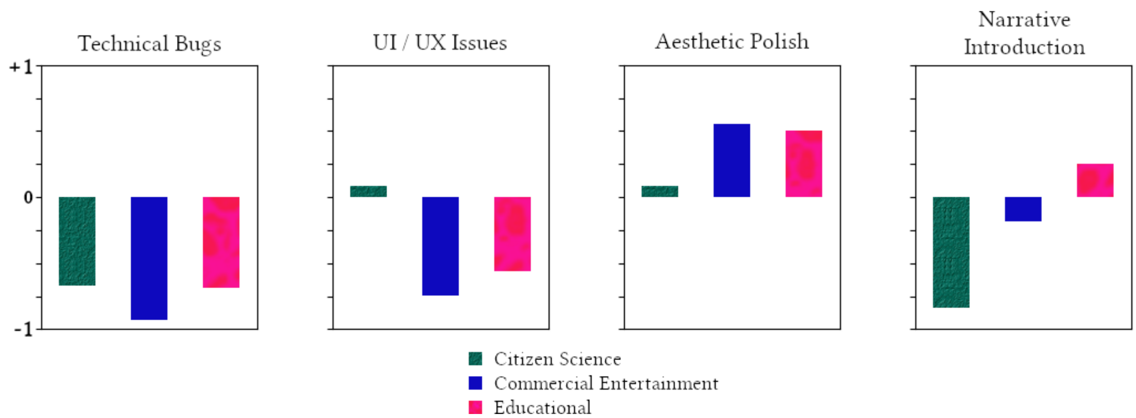


Figure 6.3: Game camps versus polish features. The Y-axis represents the average code for each camp, such that a positive instance of the feature is 1, a negative instance is -1, and a neutral or uncoded instance is 0; bugs and issues were similarly coded as 'None' (-1), 'Minor' (0), or 'Major' (1); each bar is the mean of all games in that camp. Commercial entertainment games had the fewest technical bugs, while CSGs frequently had minor or major UI/UX issues and a lack of aesthetic polish (e.g., a lack of background music). CSGs almost never narratively introduced the game prior to the tutorial, while this was a common design pattern in commercial entertainment and educational games.

minor issues with art and music. However, no game in our dataset exemplifies this case — games with aesthetic issues were always accompanied by larger, cognitive onboarding issues. Games with less polish often had technical bugs, non-standard controls, and had no way to step backward in the tutorial if one missed something (notably, this was only ever noticed in CSGs and educational games).

Therefore, we conclude that polish comes second to playability. Empirically, games with less polish are not significantly worse because of it — this was true both in our study and found by Andersen et al. [12], who saw little to no effect of music and sound effects on player retention. Moreover, Andersen et al. [12] found that animations increased engagement, which they interpret as providing feedback. In this way, their finding agrees with this theme that cognitive support is more important than aesthetic polish. We extend their results by adding that, in this study, a lack of polish was an indicator of larger issues with the onboarding design. So although music, sound effects, and so forth are not themselves critical for engagement, their presence may correlate with other, more effective matters of polish.

6.4 Discussion

In this study, we performed a close play of 55 games' onboarding experiences to understand trends in tutorial design across citizen science games, commercial entertainment games, and educational games as representative of three distinct camps or industries of game development. We generated three themes which summarize our experiences and analysis.

First, *Successful Games Pace Learning and Check Understanding*: we noticed that CSGs more often introduce mechanic after mechanic in a separated tutorial, while commercial entertainment games integrated their onboarding into the main game experience. Their onboarding gradually increased in complexity while providing clear scoring, clear feedback, and competence gates to prevent breakdowns of understanding [227, 228]. Having a separate tutorial is (as Andersen et al. [14] identified) more useful for games with complicated mechanics (e.g., most CSGs). This difference in mechanical complexity may explain why CSGs lean toward having an explicit tutorial; yet, learning science research suggests that CSGs would benefit from more spacing between the introduction of new material to allow for more practice [71].

During play, when we failed to understand a CSG, it wasn't because a new mechanic directly confused us, but because it muddied our strategy (where one existed) or allowed us to misinterpret its meaning. Entertainment games, on the other hand, were largely better about preventing misinterpretations through the use of competence gates (and several forms of feedback which are widely recommended in previous literature [471, 55, 304]). Outside of the games industry [563], however, competence gates appear to be an unknown or undiscussed design pattern. Therefore, we encourage tutorial developers to give careful consideration to challenges which test a player's mental model, challenging every assumption and misinterpretation the player could make and forcing them to consider aspects of the game that they would otherwise overlook [259].

Second, *Successful Games Set Expectations*: when we played CSGs, we had little sense of how our gameplay contributed to science — an issue raised in prior literature [332, 333]. Similarly with educational games, we entered the game with expectations about the educational value and were often disappointed. While this issue is partly addressed by clear advertising (cf. "giv[ing] the big picture up front" in the next chapter [334]), we believe it is also a matter of setting the game's scope appropriately. For example, a small indie game is unlikely to be able to cover everything a learner might want to know about a new language or biological subjects like genetics and ecosystems, but a more narrow focus (vocabulary rehearsal, Punnett squares, food chains) could be delivered effectively. To this point, we recommend using the framework of pedagogy, andragogy, and heutagogy [46]. In this framework, pedagogy means teaching novice learners through hand-holding. This is a good approach for games who intend to reach non-gamers (such as CSGs and educational games) — rather than assuming gameplay literacy, the game walks the player through basic operations gently and gradually. Andragogy focuses on intermediate learners who need a more loose hand-holding while still having guidance available; this approach works well for people familiar with games and is the most common approach to design in commercial entertainment games. Finally, in heutagogy, learners are self-determined and self-driven. This is the most open-ended and advanced learning and may be appropriate for some educational games. However, framing is critical: our mismatched expectations could be described as expecting one of these learning frames and receiving a different one. Games, especially educational games, should be clear about how much prior knowledge is expected of the learner/player and how open or advanced the game contents are.

Third, *Polish Means Playable Not Pretty*: we found that CSGs had more technical bugs and UI/UX issues and were playable on fewer platforms than commercial entertainment and educational games. Moreover, despite the recent literature promising the effectiveness of narratives in CSGs [407, 327, 330], CSGs had comparatively very few narratives. This theme is an extension of Chapter 5 which found that CSGs lack polish; here, we articulate exactly in what ways they lack polish (i.e., with respect to playability).

A game's bugs and UI/UX issues can be seen as violations of playability heuristics [161]; developers can use these heuristics (or playtest their games, or use several other methods of games user research [351, 570]) to check their software for issues. Yet, major bugs aside, the cognitive aspects of onboarding were often more frustrating than the perceptual aspects — given enough time and practice, we were able to infer what most UI elements were, even the esoteric ones, but we could not figure out the core gameplay loop without the game's assistance. From this, we conclude that polishing the onboarding design should focus on predicting and answering players' questions — making the core gameplay loop and interaction structure very clear. For example, gameplay data can be logged and mined as part of the playtesting/feedback loop of iterative development [76]; we believe, though, that frequent playtesting may be more efficient, and the games industry has good recommendations for successful playtest design [119].

Reflecting on our study more broadly, we note two points of interest that were not covered by our themes but emerged from the critical, "Big Q" approach [262] of the close reading method.

First, a major factor in the success of commercial entertainment games' onboarding is that entertainment games have, largely, simpler game mechanics. They introduce a few core concepts and much of their gameplay is about combining small strategies into complex insights in a closed system. The obvious issue for CSGs and educational games is that they are grounded in real-life concepts with a pre-determined complexity. Yet, there is a spectrum of abstraction in games. For example, in *Legend of Zelda: Breath of the Wild* [362], creating an apple pie is as simple as combining the ingredients (apple, butter, sugar, and wheat) in your inventory UI at a cooking fire; in *PlateUp!* [236] (a game which more closely simulates cooking), the player must wet flour to make dough, knead the dough into pie crust, parbake the crust, chop an apple, and bake the chopped apple in the pie crust. Both games are grounded in the real-life concept of baking an apple pie, but chose different levels of complexity in their simulations.

In this way, we argue that there is value for CSGs and educational games to simplify their abstractions and reduce their scope from early in the design process rather than creating extensive tutorials for a more detailed simulation than the development team can realistically support. Even in the most complex entertainment example for our dataset, *Kerbal Space Program* [485], many of the details of engineering a rocket are simplified to pre-determined parts that the player can combine at a conceptually simple level of abstraction — a simple rocket can be made with a rocket booster, a command module, and a parachute. In this way, the player can — from the first few minutes of gameplay — make sense of the core gameplay loop (building and flying rockets) before being exposed gradually to more complex concepts (orbital mechanics, docking ships together, and so on).

Second, educational games found success in providing learning as the reward for play — these games slowly fed the player information as the player progressed through the game, framed in such a way that the learning itself was the reward for progression (just as badges or score would be in a more traditional game). The most straightforward example of this idea was *Sokobond* [8], which provided enjoyable (albeit forgettable) chemistry facts after completing each level. Perhaps a better example though is *Tyto Ecology* [232], which includes a wealth of information about the plants and animals that the player unlocks through gameplay. Some games were unsuccessful in providing learning, the most egregious examples being CSGs that failed to explain the scientific nature of their gameplay (e.g. *Quantum Minds* [462], *Apetopia* [530]). Endogenous design (or intrinsic integration [200], where the learning goals are fully integrated with the game design and learning happens solely by virtue of playing the game [25, 270]) is the gold standard for educational games. Yet, this idea of providing learning as a reward stands as a sort of middle ground between a fully integrated design (which might be difficult for novice designers to achieve) and the often discouraged "chocolate-covered broccoli" style of design (where learning is rote, boring, and blunt, but where gameplay elements are added in order to make the experience somewhat palatable [134, 238, 573]).

How do our findings compare to previous literature on tutorial design? On common topics like feedback and cognitive load, our work agrees with prior conclusions, such as the value of positive cognitive feedback, just-in-time instruction, minimal text, fading scaffolding, and tooltips [471, 55, 169, 14]. The most similar studies that could be compared are the ones which produced a full set of onboarding heuristics. Yet, these are not well-suited for direct comparison. The Game Approachability Principles [130], for example, are not detailed in published work — the authors summarize the major categories (such as "practice," "demonstration of type," and "sandbox") but their descriptions are insufficient for meaningful comparison (aside from the authors' statement that it is important to have varying tasks and clear goals, which our findings agree with).

Similarly, when Thomsen et al. created mobile game onboarding heuristics [512], they produced heuristics such as: have a clear goal, provide player autonomy, match the player's skill level, satisfy relatedness, provide clear feedback and progress, allow for playing in varying game contexts, enable players to learn the game quickly, use music appropriately, have valuable and purposeful rewards, and immerse the player. Although we found little evidence supporting the heuristics of relatedness, varying contexts, music, and immersion, our findings agree with the rest of Thomsen et al.'s set, and places further emphasis on particular features which enable learning, such as competence gates to check understanding and setting expectations appropriately.

A more meaningful comparison may be found in examining the recent work of Poretski and Tang [396] who, similar to our methods, analyzed a variety of game playthroughs. Unlike our study, though, Poretski and Tang focused on highly popular AAA games and only watched playthroughs rather than playing the games themselves. They divided their patterns into before, during, and after (or adjacent to) gameplay.

Their "before" patterns of "recaps," and "assessing prior knowledge" resonate with our theme of checking understanding — we agree that the player should be clear about what they should be expected to know or learn. Moreover, "seeding in the cutscene" aligns with our recommendation to set expectations appropriately. These three patterns together are about clarity in what the player knows or should know before gameplay and what further gameplay will be about. Lastly, their pattern of (explicit) "tutorials" is, in our study, divided into several mechanisms. We observed that tutorials are sometimes "one pagers" (for simple arcade games), separated from the main game (especially in CSGs), and often teach concepts in this order: camera controls, other game controls, basic interactions, and finally core mechanics, in line with the practice of gradually increasing complexity.

During gameplay, Poretski and Tang observed an "invisible hand" and "sixth sense" guiding the player. These align with our code of “level or environment design.” Similarly, their “practice in a sandbox” echoes our “good pacing” and the reverse code of “mechanic after mechanic”. Their “just-in-time reminders” matches our “just-in-time tutorials”, and “personal advisers” matches “NPC mentors”. Finally, in their after/adjacent category, Poretski and Tang describe “debriefing” for feedback on the player’s performance (we refer directly to “performance benchmarks”) and “documentation” (our “on-demand info”). Through discussion of these overlapping observations, our work validates and expands on the work of Poretski and Tang. Taken together, these patterns can provide a more comprehensive look at onboarding design patterns. Notably, our pattern set includes several additional insights into *how* onboarding happens successfully, such as gradual complexity, signaling, gestures, dynamic help, and tooltips.

6.4.1 Limitations and Future Work

This study is not without limitations though. First, the researchers had some prior familiarity with some of the games played (for details, see `https://osf.io/ut4mg/`). Second, due to our selection process, the division between commercial entertainment and (commercial) educational games was muddied. In our analysis, educational games rated similarly to entertainment games on many features, and we expect this similarity is due to how we collected them, i.e., from Steam — a commercial platform. We may have seen different results if our selection of educational games expanded to classroom games, edutainment CD-ROMs, online gamified education, and other forms of educational gameplay. The educational games we selected could be considered “mass-market” education (cf. [171]).

For future work, this study prompts a more thorough examination of the instructional design in games’ onboarding. For example, the difference between entertainment and educational games could be examined through the lens of learning science principles [272] to identify which learning science principles each development camp is applying well, poorly, or not at all.

6.5 Conclusion

In this study, we examined 55 citizen science, educational, and entertainment games through a close play methodology to understand similarities and differences in onboarding design approaches across these development camps. We found that successful games pace learning, check understanding, and set expectations. CSGs suffer from a lack of explanation and playability, and educational games (at least commercially available ones) often under-deliver on their educational value expectations. Moreover, both CSGs and educational games often have more complex mechanics which could be abstracted to simpler gameplay. From this study, we emphasize the value in a gradual increase in complexity, checking understanding through competence gates, and simplifying mechanics. We recommend researchers further examine the instructional design of onboarding in games through the lens of learning science principles [272].

6.5.1 Takeaways

This chapter identified particular features that make CSGs onboarding unique from other "camps" of development. Furthermore, it gave us a few mechanical ideas for improving the onboarding design of ECCSGs. In particular, this study would suggest that ECCSGs should:

1. Integrate the tutorial and gameplay
2. More gradually increase the complexity
3. Use competence gates to check for understanding
4. Provide additional practice between new mechanics
5. Provide more explanation of the game's scientific purpose
6. Make clear what to focus on (i.e., where to allocate cognitive effort) during learning
7. Address technical bugs and UI/UX issues
8. Have a narrative introduction to invite the player into the game and prime their experience
9. Simplify the game's mechanical abstractions

A major goal of this study was to identify what makes ECCSG onboarding unique, and what ECCSGs can learn from successful onboarding designs. From these findings, I conclude that what makes them unique is the sheer number of mechanics they try to introduce before the core gameplay loop is even accessible. That is, in a game like *Kerbal Space Program*, even though it has a lot of mechanics and a standalone tutorial, one can enter the core gameplay loop fairly early — in a matter of minutes. Contrast this with *Foldit* and *Eterna*, whose core gameplay (the scientific puzzles) are completely incomprehensible until hours into the tutorial.

This is an unfortunate finding for existing ECCSGs because it implies that there is a fundamental problem with how they have designed and abstracted their mechanics — to fix the problem of too many mechanics could require an entire revision of the game. A more gentle approach would be to integrate the tutorial into the core loop and more gradually increase complexity by providing additional practice and competence gates between tutorials. However, this too is challenging to design because ECCSGs have particular scientific problems that they are trying to solve which may not be suitable as "practice." As a potential solution, these larger scientific problems could be broken up into manageable chunks that *can* be approached by more novice players in a truly crowdsourced way; I leave this as future work for ECCSGs designers to investigate.

Now that we have identified what makes ECCSG onboarding unique and how it can be improved, let us begin thinking about implementing these findings into a real ECCSG for empirical testing. In the next chapter, I prepare for implementing a new tutorial approach in *Foldit* in two steps. First, I consider the instructional design of *Foldit* (and other ECCSGs) through the lens of *skill chains*, which are a useful concept for gradually scaffolding mechanics and their dependencies. In doing so, I investigate how players and developers conceptualize skill chains to determine if players and developers are thinking about their skills in the same way and whether one or both of their mental models could be a useful map for designing a new tutorial. Second, I perform a skill-based cognitive task analysis on *Foldit* to more thoroughly document the skills which *Foldit* requires for expert play, which finally sets up Chapter 8 for implementation.

Chapter 7

Instructional Design of ECCSG Skills

Suppose you were to design the tutorial for an ECCSG from scratch: how would you start? One possibility is to map out the mechanics you need to teach — or reversed, the skills that the player needs to learn. You could diagram these skills hierarchically, since some skills must be learned before others. Not only is this an intuitive approach, skill mapping is backed both by instructional design theory (e.g., [537]) and industry game development [95]. Therefore, in this dissertation, I employ veteran game designer Daniel Cook's *skill chain* framework (which has since been adopted by games research [134]) to model ECCSG onboarding design [95].

In 2007, Cook coined the phrase *skill atom* to refer to an atomic player skill and *skill chain* as a hierarchical list of skill atoms, such that later skills require prior skills [95]. As a brief tangent for disambiguation, the term skill chain is closely related to a *skill tree*, which is a game mechanic in roleplaying games that allows the player to customize their character's abilities by selecting from branching options of ability unlocks or upgrades. A skill tree (which might be more appropriately thought of as an ability tree or upgrade tree) differs from a skill chain in that the nodes refer to character abilities rather than player skills. Related to the skill tree is the "tech tree" (sometimes research tree) of 4X games in which players upgrade the knowledge capital, such as science and technology, of their faction over time. In this way, skill *trees* model the *game state* while skill *chain*s model the *player*. Given the usage of "chain" as opposed to a skill "tree," one might think that a chain is non-branching; however, this has not been the usage of the term in previous literature — skill chains can branch. For consistency, I also use the term "skill chain" to refer to the entire composition of player skills, despite the fact that it branches like a tree.

Skill chains can serve three purposes. First, they provide an outline for what skills need to be introduced during onboarding and in what order based on skill dependencies. Second, they

‡Parts of this chapter were adapted from [334].

can help developers identify breakdowns [228] during playtesting by enabling developers to isolate where in the chain of skills the player is failing to progress. Finally, skill chains can be useful as a player-facing tool to help them track and understand their progress. For example, Tondello and Nacke applied a skill chain to gameful education to help students recognize and practice their skills [520].

Yet, this raises the question: is a skill chain primarily for describing expertise or the path to expertise? In a sense, it is both, since one who learns and applies the skills used by experts (and all of their prerequisites) will almost by definition become an expert theirself. A skill chain is both the curriculum and the job requirements: it describes what learners need to know as well as what skills are used by experts. By modeling what the player needs to know via the development of a skill chain, CSG designers can better create effective tutorials and onboarding systems. This, in turn, better prepares the player for the CSG tasks, which improves retention as well as data quality and quantity of the scientific output.

However, determining the skill chain of a game is an arduous process. Previous work attempting to extract knowledge about the skill chain of a CSG through Cognitive Task Analysis proved to be an intensive task [218]. A much simpler, potentially cheaper strategy to develop a skill chain would be directly asking the expert players to draw the skill chain as they understand it. This direct, unguided approach is typically avoided in CTA literature because it can lead to reduced or less structured results [84]. However, two critical factors differ in this context. First, rather than an interview or task diagram, the output we are looking for is a skill chain, which may inherently structure the problem for the players. Second, video games are more structured than typical domains: for example, the game is divided into levels, has an explicit tutorial, and has an explicit goal with immediate feedback.

This line of thinking prompted my co-authors and I to develop a study on skill chains. We tested the efficacy of freely recalling skills in this context in the hopes that this method would be an easy but effective means of developing a skill chain. Moreover, in pursuit of designing an ECCSG tutorial myself, this study serves as preparatory work to understanding how players conceptualize the skills they are learning in ECCSGs.

While considering how players view the skill chain, we were also curious how developers view the skill chain. Are developers' and players' understandings complementary, able to be merged into a cohesive skill chain? Moreover, does the existing skill chain model accurately capture the way players and developers conceptualize skills?

The following study had two goals. First, we aimed to explore how players and developers conceptualize the skills gained through play, such as to investigate if their mental models align with each other and/or with the skill chain model. Second, we aimed to test the efficacy of free recall in the direct elicitation of skill chains for CSGs, since free recall might be more effective than usual in this novel context. We addressed the following research questions:

RQ1. How do players and developers conceptualize the skills gained through play?

RQ2. How effective is free recall as a method for directly eliciting the skill chain of a CSG from players and developers?

To answer these research questions, we elicited skill chains from 16 players and 6 developers of 3 ECCSGs: *Foldit*, *Eterna*, and *Eyewire*. We additionally member-checked [105] with 11 of the participants via semi-structured interviews to confirm our thematic analysis of the skill chains. We found that: (RQ1) players and developers conceptualized skills in four ways — tutorial-oriented, core loop, stream of thought, and WYSIATI;[1] (RQ2) direct elicitation was comparable to the efficacy of free recall in other contexts for the purpose of understanding a game's skill chain; however, it was effective for eliciting the core gameplay loop, tutorial overviews, and some expert insights from their recent gameplay experiences, which may be of value in early-stage analysis of a game and its skill chain. In this way, our method is useful for studying existing games rather than for the development of new games for which there are not yet any expert players. Given the cost of skill chain development, it is arguable (albeit, for future work) whether there is any value in considering the skill chain of a game in development, both because the skills needed might change as the game grows and because expert players will inevitably use different skills and strategies than developers intended.

Our three main contributions of this study are: (1) a comparison of skill chain conceptualizations between players and developers and across prior literature (i.e., the skill chain models of Cook [95] and Deterding [134]); (2) insights to the process of free recall in eliciting ECCSG skill chains; and (3) a preliminary toolkit of ECCSG skill-based design recommendations based on our findings.

[1] What You See Is All There Is

7.1 How do Players and Developers of CSGs Conceptualize Skill Chains?

7.1.1 Background

This work is situated within the context of *iterative playtesting* for *onboarding design* through *player design* of *skill chains*. However, much of the background on playtesting, onboarding, and player design are beyond the scope of this chapter, with one notable exception. In the games industry, developers at Jagex Limited (creators of the MMORPG *RuneScape*) have experimented with crowdsourced player design with mixed results [372]. Relevant to this work, they found that crowdsourced design has benefits to player engagement but is limited in two major ways: (1) which players are willing to engage with player design, and (2) what ideas are generated from players. These findings can be seen as potential limitations to using the current methodology for generating design ideas, a conclusion which is backed by the results of this study as well.

7.1.1.1 Cognitive Task Analysis in Games

Broadly speaking, this study is a CTA, of which there are many varieties of methods and purposes [103]. Specifically, we focus on analyzing how one gains expertise at a particular (game) task. The most similar study was conducted recently by Hesketh and Deterding who applied grounded theory to investigate how novice to intermediate players gain expertise in team-vs-team esports games [213]. Their findings highlighted three aspects of gaining expertise in games: learning processes (identifying knowledge/skill gaps, consuming and internalizing information, applying knowledge/skills in new contexts or combinations, and practicing knowledge/skills), learning tools (game modes, add-ons/extensions, streaming services, forums and other communication channels, and statistics services), and learning goals (basic controls, game mechanics, motor skills, strategies, game-meta, non-game-specific knowledge/skills, and meta-learning skills).

7.1.1.2 Expertise and Expertise Modeling

Much has been researched about the cognitive differences between experts and novices. In chess, for example, experts are able to more quickly memorize board positions by chunking relations using cognitive schemata they have learned over time; expert chunks were empirically found to capture more data, and some evidence suggests they may have been able to retain more chunks in memory [73]. Building on this "perceptual chunking" hypothesis, researchers have studied how expert and novice physics students create problem representations of physics problems [78]. They found that experts' representations are more abstracted in the principles of the domain — including schemata for solution methods — while novices rely on surface features of problems.

In HCI, further work has been done to characterize mental models, such as the Knowledge Component (KC) model [9]. In this framework, a KC represents a cognitive unit, such as a fact or procedural skill. Related to the current study, Harpstead and Aleven have applied the KC model to educational games in order to empirically analyze the learning curve of a game [9]. Using this method, they were able to help the designers refine the game based on new understandings about the skills players were using; however, the authors note that this method requires a large amount of player data for the statistical techniques employed. Notably, prior to Harpstead and Aleven's study, no research has looked at identifying or correcting designers' potential misconceptions about the skills they believe are used in their game [9]. The present study thus builds on Harpstead and Aleven's work to continue looking for ways to refine the designers' model of player skills. We explore direct elicitation of skill chains as a potential method for identifying these misconceptions without the high cost of large amounts of player data.

There have also been studies on expertise in games specifically, such as on the skills used by professional esports players [158] (and, on the other hand, learned by novices [213]), social and metagame aspects of expertise [141], learning curves and the habits of experts [223], and the markers of "extreme expertise" [300]. However, few of the previous works have looked at learning or expertise for the purpose of skill modeling or tutorial refinement, opting instead to be descriptive of what expertise is or how learning happens. The present study aims to produce both descriptive and prescriptive results, enabling CSG designers to improve their existing onboarding structures by unpacking how players and developers mentally model the game and how knowing those mental models can influence better tutorial design.

7.1.1.3 Skill Chains and Skill Modeling

In Cook's *skill chain* model [95], each *skill atom* consisted of a player action which leads to a system response (simulation) followed by feedback on how the system state has changed, ultimately resulting in the player updating their mental model of their interactions with the game. Cook further adds that skills can be mastered, partially mastered, unexercised, active, or "burnt out" (i.e., the player becomes disinterested in exploring further skills built on this atom). More "advanced elements" of the skill chain include *pre-existing skills*, those gained prior to beginning the game, and *red herrings*, which "will never result in a useful in-game skill, but ... still evokes the pleasure of partial mastery in the player" [95].

In 2015, Deterding built on Cook's model with the additional consideration of the player's motivation [134], which is especially practical in the consideration of serious game design. Deterding's list of skill atom components included (player) goals, (player) actions, (game) objects, (game) rules, (game) feedback, (game) challenge, and (player) motivation.

Both Cook and Deterding consider the atomic skill to be an action. When this assumption is held, skill chains become algorithmic, and in fact previous work has attempted to apply skill-based algorithms consisting of atomic actions as a way to automatically playtest the skills required in a game [219]. However, in the realm of player behavior modeling, skills can be actions, tactics, or strategies [30]. Bakkes, Spronck, and van Lankveld define tactics as "short-term/logical game behaviour as composed of a series of game actions," while strategies are "long-term/global game behaviour as composed of a series of game tactics" which may span the entire game or multiple games [30].

Skill chain theory agrees with the model matching theory of McGloin, Wasserman, and Boyan [54]. They write that the process of improving one's play involves *matching*, or "the extent and accuracy of alignment of a player's mental models with a game's constellation of mechanics." The term "constellation of mechanics" quite explicitly evokes the representation of the skill chain model, suggesting that expertise is the process of discovering, exploring, and grokking (or deeply understanding) each mechanic in the skill chain. One key idea of model matching theory is that players iteratively refine their own mental model to match the game model, a proposition which has gained recent evidence [554]. This suggests that players have their own conceptualization of the skill chain, perhaps implicitly, which they use to map their understanding of the affordances and constraints of the game and inform decision-making [54].

Some work on educational games has focused on *knowledge tracing*, which is effectively measuring the mastery of each skill atom in a skill chain for a particular player based on performance [251]. Kantharaju et al. [251] define the skill chain using the CTA methodology of Horn, Cooper, and Deterding [218] (discussed below) and define successful skill usage through binary behavior metrics. For example, the skill "Testing before submitting" is defined by the behavior "Player tests before submitting."

Another study operationalizes Cook's skill atom theory to implement Talin, a dynamic tutorial framework in the Unity game engine [26]. In this framework, each skill atom in the skill chain (manually defined by designers) holds a *mastery* scalar. The designers then add *detectors* to the game world which detect opportunities to use skills: while a player doesn't take this opportunity to use the skill, mastery decays. If, instead, the player exercises the skill, mastery increases. Detectors can also trigger predetermined hints (such as pop-up text or visual cues) dynamically based on the mastery values.

Within serious gaming, a new dimension of skill is introduced. Game mechanics are not always perfectly integrated with the serious mechanics (e.g. educational content, human computation tasks, or citizen science tasks; cf. [200, 22]). For this reason, Sarkar and Cooper develop a disjoint skill model for simultaneously tracking game skills and task skills [448]. Through this disjoint modeling, they were able to introduce players to overall more difficult tasks via more nuanced dynamic difficulty adjustment (DDA).

Most prior work on skill chains has assumed the designers can manually generate the skill chain. However, this process doesn't take into account empirical player behavior. In his 2019 dissertation on skill chains, Horn writes:

> *"Scarcely any game research methods exist to empirically deduce the skill chain of a game from actual player experience, assess to what extent the skills and ideal sequencing order predicted by a model matches the skills it requires from players, assess the efficient acquisition of those skills by players, or the optimal learning hierarchy. This risks overlooking essential skills, not introducing them to players, or introducing them in a sub-optimal sequence."* [217]

To address this, Horn, Cooper, and Deterding attempted to elicit skill chains via (Skill-Based) Cognitive Task Analysis (*SB*CTA) [218]. This process involved semi-structured interviews with video-aided recall of play sessions. Interviewers attempted to "elicit procedural and automated knowledge around low-level gameplay as well as representational decision-making and strategy skills." Through analysis, the authors make six relevant and generalizable conclusions: (1) novices were more valuable than experts in identifying low-level interface and gameplay skills, (2) skill dependencies are unclear and confounded by level design, (3) the distinction between procedural and strategy skills is fuzzy, (4) skill chain analysis surfaces low-level and pre-existing skills, (5) skill chains run together in a core mechanic, and (6) skill chains remain flat. Ultimately, the authors conclude that the CTA methodology produced something too raw and unstructured for practical use (personal communication, Seth Cooper, 2019), making it difficult to merge the results into a usable skill chain [218].

Though beyond the scope of this chapter, skill modeling also sees representation in Intelligent Tutoring Systems (ITS), such as via Bayesian Networks (BN), Case-Based Reasoning (CBR), and Partial Ordering Knowledge Systems (POKS) [217, 187, 126]. This has been applied, for example, to automatically generate a partial ordering of practice problems for language learning [553].

However, this process requires already knowing the skills to be introduced. This study focuses on the previous step: understanding the skills and dependencies involved in the task.

Cowley and Charles offer another means of conceptualizing player modeling of game actions: “Behavlets” [102]. In short, rather than dividing behavior into the atomic skills to be learned and applied, Behavlets capture the atomic player behaviors of a game: the significant, psychologically-informative features of gameplay. However, Cowley and Charles mean to use this model as a way of analyzing player traits and codifying player behaviors, rather than understanding expertise or the progression thereof. Future work may be interested in combining these approaches by using codified atomic behaviors as a means of measuring skill usage and mastery.

Lastly, skill chains have been applied within *Foldit* specifically [219]. Horn et al. produced AI “Stratabots” which attempted to complete the tutorial levels using only certain skills, thus validating whether the tutorial levels in fact taught the skills that the developers meant to require of the player. Notably, the authors found that using only a couple of basic game mechanics, a player is able to complete many tutorial levels without needing more advanced skills, which suggests some inefficacy in *Foldit*’s current tutorial levels.

Therefore, the current study is a direct extension of prior work eliciting skill chains from players. In an effort to find a scalable but empirical way to determine skill chains, Horn et al. [218] attempted CTA but found it too intensive as a method. Early knowledge elicitation methods suggest that “the most direct way to find out what someone knows is to ask them” [96]. Thus, the current study takes a more direct approach: rather than eliciting the skills via interviews with video-aided recall, can we simply ask the players directly what the skills and skill dependencies are?

In this way, our method of direct elicitation is related to free recall. Although unaided free recall has been shown to produce only 30% of an expert’s knowledge (cf. the 70% rule [86]), we hypothesized that eliciting video game skills may be a different enough context for this rule to no longer apply. That is, because video games are more explicit than typical CTA contexts about the skills an expert learns — such as by dividing the game into levels, having explicit tutorials, and providing more overall structure to the task and user interactions — we hypothesized that video game skills may be easier to elicit directly than skills from other domains of expertise. However, as detailed in the Discussion, this hypothesis was not supported.

7.1.2 Methods

The three games examined were *Foldit* [531], *Eterna* [487], and *Eyewire* [408], chosen to be a representative sample of ECCSGs. For a description of these games, refer to Appendix A. Methods were approved by Northeastern University's Institutional Review Board.

7.1.2.1 Participants

Players (16) were recruited through purposive sampling. Expert players (12) — and one *Foldit* novice — were contacted online via in-game messaging systems and game forums. The remaining three *Foldit* novices were recruited from university students via online mailing and messaging lists, e.g. Slack and Discord, and screened to be at least 18 years old and without ever having played *Foldit*. To protect the anonymity of our participants (and because we don't expect expertise to be affected by these variables) no other demographic data were collected, such as age or gender. However, based on information that participants have publicly released (such as on their profile pages within the game), we believe our participants to be an accurate representation of the ECCSG player population as described by Curtis [115] in her synthesis of 13 studies containing demographics of ECCSG players: participation is biased strongly toward older, Western, well-educated males from developed countries (primarily within the U.S. and Europe), with disproportionate biases toward IT-related professions. Although we do not have specific data on all of our participants, our sample is — to our knowledge — reflective of the known demographics of ECCSG players.

We also did not ask participants for a specific description of their level of expertise, since we expected our sample to be too small for these data to be meaningful. However, member-checking interviews and public information from player profiles (where available) revealed that experts tended to range in experience anywhere from 1.5 to 11 years, varying as well with the age of the game itself — experts of *Foldit*, the oldest game, tended to have at least 5 years of experience.

Players were offered a $15 USD Amazon gift card as remuneration. Developers (6) consisted of collaborators on this project and had the option to contribute to and be co-authors or acknowledgements on this work. The primary analysis in this work was carried out by one of the *Foldit* developers who is an author. As a form of member-checking [105], 11 participants were interviewed in a semi-structured format about their skill chain and play experiences.

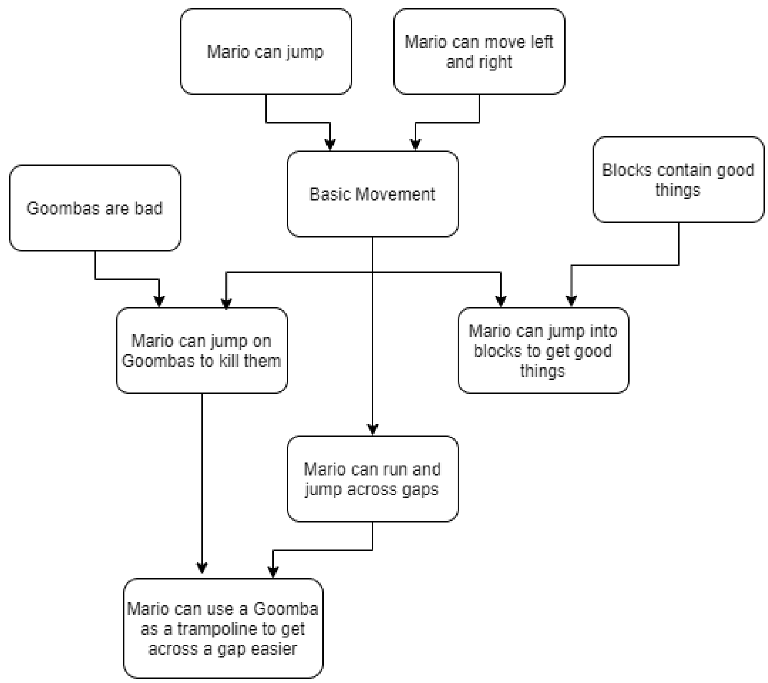


Figure 7.1: The example skill chain shown to players. This simple chain was constructed by the researchers for the purpose of example.

Game	**Novice players**	**Expert players**	**Developers**
Foldit	4 (1)	6 (3)	2
Eyewire	0	3 (1)	1
Eterna	0	3[†] (3)	3[‡] (3)
Total	4 (1)	12 (7)	6 (3)

[†] One player (P16) submitted two skill chains; we count this as one chain but refer to them as P16a and P16b.

[‡] All three developers of *Eterna* were first avid *Eterna* players before becoming developers, granting them a unique perspective to the game's workings.

Table 7.1: A summary of participants and data in the study. The table lists a count of skill chains; a count of member-checking interviews are listed in parentheses. In total, the study involved 16 players, 6 developers, 23 skill chains, and 11 semi-structured member-checking interviews.

7.1.2.2 Procedure

Skill Chains Expert players and developers were referred to Cook's Game Developer article[2] on skill chains (including his Tetris example) and an example skill chain of Mario (see Figure 7.1). Then they were asked to generate their own chain, such as by using the diagram-making tool *draw.io*,[3] although some participants opted to submit a plaintext file, spreadsheet, or diagram made with other software. Three *Foldit* novices followed a similar procedure in-person, but before generating a skill chain they were brought to a computer in a quiet, comfortable room to play the *Foldit* tutorial for 30 minutes followed by a science puzzle for 10 minutes.

Interviews Experts, developers, and the *Foldit* novice recruited online were contacted for interviews as described in Section 7.1.2.1. Interviews were semi-structured with the intent to elicit insights about the cognitive process of making their skill chain and other conceptualizations of skills, such as what their first and most recent skills learned were and how they might re-imagine the game's tutorial to better teach the skills necessary for expert play. Each interview lasted approximately one hour.

Data Collected As shown in Table 7.1, our final dataset consisted of 23 skill chains from 16 players and 6 developers, as well as 11 interviews (approximately 11 hours of transcribed audio) from 8 of the players and 3 of the developers.

7.1.2.3 Analysis

We first performed a multi-coder codebook thematic analysis [61, 58] on the 23 skill chains specifically to answer the research question "How do players and developers conceptualize the skills needed to play ECCSGs?". This work is theoretically framed from a constructionist perspective that assumes people create mental models to understand the world around them [460]. Moreover, we acknowledge that we also bring in assumptions about how learning happens through games as transmitted by the mass culture of gaming and game tutorials.

This theoretical approach led us to a deductive analysis driven by (1) our research questions and (2) constructionist theories of learning. We additionally took a critical orientation to sense-making for this analysis. However, we include an element of critical realism to our approach in that we were open to the data providing evidence against our assumption that their experiences

[2] https://www.gamedeveloper.com/design/the-chemistry-of-game-design
[3] https://app.diagrams.net/

are mentally modeled in this way. Therefore, we code both for semantic and latent meaning in order to capture the player and developer experiences as well as how these experiences might be interpreted through the lens of constructionist learning, skill modeling, and common game design theory [95, 134, 451].

For a preliminary, exploratory analysis, the generated skill chains were iteratively coded by three coders.[4] The coders independently familiarized themselves with and coded the data according to the research question, holding the assumption that each node (i.e., atomic skill) of the skill chains produced would be given exactly one label from a set of categories (i.e., node/skill types, cf. [30]). The coders then convened to discuss their labels and revise the shared category set based on (1) relevance to the dataset as a whole, (2) generality where applicable, and (3) specificity (avoiding "bucket" labels), with the assumption being that each category should be significantly and generally representative within the data but distinct from other categories. After five rounds of iteration the categories stabilized, at which point the primary coder performed the remaining analysis following Braun and Clarke's reflexive methodology [59, 58], generating individual codes which were then aggregated into themes.

This second analysis, performed by the primary coder only, dropped the assumption that each node would have exactly one label. Instead, it examined two perspectives. First, when holding the assumption that players and developers conceptualize skills as proposed by the skill chain model, how do the defined categories help interpret the skill chains elicited? Second, when freed of the assumption that players and developers rely on an underlying skill chain model, in what ways do they actually conceptualize their skills and game learning experiences? This analysis took four additional iterations through the data and resulted in a total of 164 unique codes which were then aggregated into 18 distinct sub-themes across 4 major themes. These themes were then revised using the interview transcripts to guide the final phase of analysis, i.e., to ensure that the themes were consistent with participants' reported experiences in the interviews.

For validation of our methodology and future research, anonymized skill chains and an analysis audit trail[5] are available at: `https://osf.io/4evfk/`.

[4] One skill chain was provided by the primary coder as a developer. The other two coders were neither players nor developers.

[5] For privacy, please contact me directly regarding access to anonymized interview transcripts.

Category	Description	Example*
Actions	Any tool or in-game ability the player can use; a single player input	Freeze
Practice	The act of repetition or continual play, especially with a focused goal, often to sharpen soft skills in the game	Practice coloring
Procedures	A specific sequence of actions or a combination of inputs	Rubber bands pull sheets together
Strategies	A high-level plan for reaching a goal; unlike Procedures, a Strategy does not specify a particular sequence but instead provides heuristics and guidelines such as if/then statements with freedom regarding how the goals are to be executed; this category also includes specific decisions made during the strategizing process	Hand fold[ing]
Guidance	Any instruction or assistance provided to a player, either from the game (such as tutorials, feedback, tooltips, and paratexts such as wikis), or between players (such as mentorship)	Press W to wiggle the protein
Discoveries	Any observations which affect the player's mental model, such as learning new game rules, experiencing epiphanies about the effectiveness of different strategies, or noticing informative details in the game state	Clashes ... are bad
Social	Collaborations, competitions, or communications with other players	Competition is fun
Objects	Any specific game element, resources, or in-game entities and concepts	Score

Motivation	Any goal, reward, or other motivating factor that the player considers; these are combined because the player uses motivation to inform a goal which leads to a reward which satisfies the motivation in a continuous, repeating cycle	Rank seeking

Table 7.2: Label categories produced by three coders via codebook thematic analysis: nine conceptualizations of the skill process with examples from the data. *All examples are from *Foldit* skill chains except “Practice coloring” from *Eyewire*.

7.1.3 Results

When maintaining the assumption that players and developers would generate a skill chain, the codebook analysis yielded nine conceptualizations — or categories of nodes — that existed in the diagrams of players and developers modeling their game: Actions, Practice, Procedures, Strategies, Guidance, Discoveries, Social, Objects, and Motivation (see Table 7.2 for descriptions and examples).

Several participants (notably, only expert players) categorized their own nodes and provided a legend for their skill chains, which we examined as another form of member-checking the categories generated. The participant legends are compared to the generated categories in Table 7.3. All generated categories can be mapped to an item in someone’s legend, and conversely all items across all legends can be mapped to the generated categories. Therefore, we conclude that the categories are grounded in the players’ conceptualizations of categories as well. In this table, we also compare the generated categories to previously modeled skill atoms from Cook [95] and Deterding [134].

7.1.4 Themes

Following a reflexive thematic analysis approach [59, 58], the primary coder generated four major themes from the data, supported by evidence from member-checking interviews. Notably, the thematic analysis did not capture the skill knowledge itself, as a content analysis might provide. Instead, we focus on RQ1: How do players and developers conceptualize the skills gained through play? This aim of understanding skill conceptualization meant that our analysis was more

Categories of This Study	Cook's Atoms	Deterding's Atoms	P11, *Eyewire* Player Legend	P6, *Foldit* Player Legend	P9, *Foldit* Player Legend	P10, *Foldit* Player Legend
Actions	Action	Actions	Player Action	Action	Controls / Actions	Player Action
Practice	-	-	-	Investment (personal & community)	-	-
Procedures	-	-	-	-	Side Issues*	-
Strategies	-	Challenges**	Player Decision	-	-	-
Guidance	Feedback	Feedback	-	-	Definition / explanation	Veteran / Science Input
Discoveries	Modeling	Rules**	-	-	-	Player thought
Social	-	-	-	Social	-	-
Objects	Simulation	Objects	Game Information	-	Concepts	Visual Element
Motivation	-	Goals, Motivation	-	Incentives	-	-

*This item was used to define simple problem-solution mappings (e.g., how to control the camera if you can't see your protein), notably using the language of *a prototypical novice's journey to expertise*.
**From a player's perspective, the rules are discovered and strategies are developed to overcome challenges. Therefore, rules and discoveries are two sides of the same process, as are challenges and strategies.

Table 7.3: (Left) A comparison of previous skill atom models to the current category set derived from nodes used by our participants. (Right) A comparison of the label categories produced to the categories provided in legends by players and developers, as a form of member-checking. Interestingly, all legends included some notion of the player's actions, and most had a notion of game elements. All categories generated in this study can be mapped to an item in someone's legend, and conversely all items across all legends can be mapped to categories generated in this study. Therefore, our categories are necessary and sufficient for capturing both traditional skill models and diverse skill-based representations (see Section 7.1.5.1).

focused on latent structural and psychosocial features than surface content. The closest we come to the content itself comes in the first theme which describes the participants' acquisition of expertise. This theme was termed *Experts are Experiential Learners* because the experts' knowledge was framed experientially in the context of how the experts learned the game behavior they perform (i.e., through observation and gaining an expert's intuition, then applying what they absorbed). The second theme could also be considered part of a skill chain, though to a less helpful extent. This theme was termed *The Process of Playing* because it captures how participants would describe the objective, fundamental interactions of the game. This information is useful in four cases: (1) instructing new players, (2) reminding developers what is necessary to teach, (3) onboarding external developers to the pre-existing instructional design, and (4) understanding the players' expression and structure of this information (as described later, seeing the instructional design through the lens of the players' experiences may provide new insights for iterative design). The third theme captures a latent conceptualization that both players and developers consider the tutorial to be a static, objective experience affecting all players equally, termed *Tutorials as Passive and Standard*. The last theme captures the "why" and "how" of the skill chains elicited, including information given by the participants on player motivations and how the skill chains elicited were structured, termed *Knowledge Framing*.

7.1.4.1 Experts are Experiential Learners

Expert players, especially of *Foldit*, commonly described their learning process and their currently known skills. Most of their comments to this effect fell under 11 sub-themes (in italics), all of which can be summarized as experiential learning with an emphasis on observation which we refer to as "eye-and-apply."

Sometimes they would describe *how their observations led to their current behavior*, either from a past experience that enabled learning — a form of cognitive apprenticeship [93] — or an emergent behavior derived from the observation and interpretation of the game's rules and framing.

> *"I found that bad regions on the Rama Map[6] tend to stay bad, so it is important to get them nearly right early in the game."* (P7, *Foldit* Player)

> *"Attaching bands by hand seems easier when something is wiggling, so I often wiggle sidechains when attaching bands by hand."* (P7, *Foldit* Player)

[6] A tool in *Foldit* that allows players to visualize and modify specific angles of the protein's fold.

Part of this process involved *gaining an "eye"* or intuition for the decision-making, strategizing, and evaluation (cf. [395]). This included identifying conceptual elements that aren't visualized in the game, observing the meaning and interpretation of visual elements, and explaining game states and actions.

> *"Harmonious design is good."* (P6, *Foldit* Player)

> *"AFTER 5 YEARS of PLAYING Foldit I NOTICE I intuitively know rather than understand and play from how the pattern looks and feels more than from my scientific knowledge, which is apparently improving but not in conscious ways. The key to my own way of playing Foldit is how a pattern looks rather than knowing why it folded up correctly from a scientific point of view."* (P8, *Foldit* Player)

The road to expertise was dotted with *discoveries they've made* of game rules that generalize, tricks of the trade, strategies, social collaboration, and specific moments of epiphany. Often these discoveries lead to new or improved strategies. One *Eterna* player/developer (D5) had an entire section labeled "Puzzle revelations."

> *"Realizing the importance of hydrogen bonds in making good structures from B-strands. Puzzle 630...really brought this idea home."* (P7, *Foldit* Player)

> *"AHA! moment with blueprint, if structure failing ideal or scoring low, chirality is probably off and shifting sheet/loop may fix. Blueprint fixed many a crummy monomer. Another AHA! Curving sheets often make a higher scoring monomer."* (P8, *Foldit* Player)[7]

Importantly, expert players were not learning on their own. They highlight *social learning and socialization* as key components of the learning process. This included receiving guidance from others, collaborating, learning from observation, and having social strategies as well as personal strategies.

> *"Group forum... group shares... Wiki top results pictures..."* (P6, *Foldit* Player)

> *"ask for help in chat... be active in chat"* (P13, *Eyewire* Player)

> *"Modify designs of other players"* (P16b, *Eterna* Player)

[7] This quote contains a lot of specialized language for *Foldit*; however, the reader does not need to understand this jargon. The point here is the player's use of "AHA" as linguistic markers for discovery.

> *"Joined group and learned from group players about the ways you can do things... Group learning very important, and a key to my own arc in the game..."* (P8, *Foldit* Player)

> *"find players who can help you"* (P14, *Eterna* Player)

This includes sharing *community-created knowledge* such as new terms for common procedures, new procedures, and the assimilation of external background knowledge. *Eyewire* players (P11 and P13) refer to "black spills" as a term they've coined for spill-like stains in their labeling dataset. An example from *Foldit* is "Space bands (I.e. Bands to empty points in space, AKA Zero Length Bands)" (P10, *Foldit* Player).

Social learning is strongly dependent on the *use of paratexts*, such as player wikis, streams, videos, and other tutorials or guides, as well as scientific literature and other professional media describing the topic of the game. Players highlight the importance of (and reliance on) these paratexts for learning and describe applying paratextual knowledge as a strategy itself.

> *"further informations: notifications, Eyewire blog, Eyewire forum, Eyewire wiki, Eyewire museum"* (P11, *Eyewire* Player, punctuation added for clarity)

> *"The Black Belt Folding videos showed me the value of using the Selection Interface..."* (P7, *Foldit* Player)

These paratexts give expert players critical *external background knowledge*, such as scientific terms not introduced by the game, which they combine with their game knowledge to apply to their decision-making. This knowledge allows them to elaborate on their reasoning or understanding of the game and model game objects using contextual knowledge. In some cases, background knowledge becomes a prerequisite for understanding certain game concepts.

> *"Hydrophobic: 'Water hating' sidechains...are colored orange...and do not bond well with water... Hence, most proteins in solution will have hydrophobic proteins facing inwards (away from the outside aqueous environment)..."* (P9, *Foldit* Player)

> *"Other [scientific] models exist with different parameters and behaviors"* (D6, *Eterna* Player/Developer)

However, one aspect of background knowledge was considered essential, and that is *the importance of understanding "the big picture"*, or why the mechanics, dynamics, and aesthetics [225] of the game — especially the goals — are what they are in the broader context.

> *"The barebones basics of what even is a protein, what are the rules of folding, and what we should be looking for when folding... emphasizes the background knowledge needed to understand what even is "good" in this game... keeping players focused on the big picture...."* (P10, *Foldit* Player)

Yet, knowledge alone is not enough for expertise. Players and developers emphasized *the need for dedicated practice*, alluding to soft skills that need to be learned, describing skills that generalize or transfer through practice, and recounting their own trial-and-error learning.

> *"...learning by doing experience..."* (P6, *Foldit* Player)

> *"User sent to practice cube...x5"* (D3, *Eyewire* Developer)

Players also expressed *self-reflection*, evaluating the performance of tool usage and strategies and reflecting on personal preferences and common behaviors.

> *"EARLY game experience: Frustrating tutorials, crappy early beginner puzzle results, stab in the dark work on some intermediate puzzles, more frustration... Learning what works... My ED [Electron Density] skills are very poor, but I see them slowly improving..."* (P8, *Foldit* Player)

Ultimately, through a combination of social learning, practice, and reflection, the players gain the background knowledge and intuitive eye for what works, leading them to *apply situational strategies* (hence why we call this particular style of experiential learning "eye-and-apply"). Based on the situation, experts apply different visualization or gameplay settings (cf. [336]), even referring to these settings as tools of themselves. They describe a mapping of problems to solutions, often via if-then rule procedures, and describe the rules and exceptions to those rules. They identify situation-recognition as a skill and identify the range of possibility space of these situations, or describe the gradual discovery of this range. This enables them to apply higher-order thinking and planning to their decision-making.

> *"If things seem stuck (like when hand-folding), use a low clashing importance to help things move to where you want them.... Sometimes you have to accept a loss in score in order to raise the score... if you are making a major change by hand, it often helps to do some wiggling and let the score fall a bit before starting your next recipe..."* (P7, *Foldit* Player)

In the above quote, for example, the player identifies the non-intuitive strategy of doing something which results in a lower score in order to get to a part of the solution space capable of reaching an even higher score. Therefore, intuitive and "greedy" strategies to score optimization would fail without this expert situation-recognition and higher-order strategizing.

In summary, experts of ECCSGs are "eye-and-apply" experiential learners. They observe and experiment, both individually and socially, and then apply situational strategies based on the procedural knowledge and heuristics they have observed.

7.1.4.2 The Process of Playing

This theme captures how most participants tried to describe the interactions of playing the game, which is conceptually separate from the skills used to play. Although experts gave deeper descriptions and novices gave more surface descriptions, most participants paid a surprising amount of attention to the game's controls and other procedural details.

The most common code in the dataset, which became its own sub-theme, was players and developers describing *how the interaction happens at a surface level* (Figure 7.2). This includes listing the game objects, tools, and available interactions between them (describing the surface affordances), listing the game rules and overt goals observed, and describing the common low-level player input controls (those explained by the tutorial and/or used by everyone). This trend in the data was most common with developers, novice players, and *Eyewire* players.

Although *Foldit* and *Eterna* expert players also described some surface-level interactions, they more often described *how the interaction is understood at a detailed level*. This included listing procedures, strategies, and heuristics for evaluation and decision-making, as well as describing the uncommon low-level player input controls (those not explained by the tutorial and/or used by a subset of players).

> *"Secondary structure controls....Right click the restructured residues -¿ Ideal SS... Sheets will require another sheet to form hydrogen bonds (you can form one by making the protein do a hairpin and go back the other way)...For design puzzles, secondary structures can alternatively be assigned using the Blueprint tool..."* (P9, *Foldit* Player)

7.1.4.3 Tutorials as Passive and Standard

This theme captures that most participants saw onboarding as a fixed experience. Additionally, novices and developers in particular seemed to share the assumption that the tutorial was the only onboarding, thus *equating the tutorial with the onboarding experience*. This included

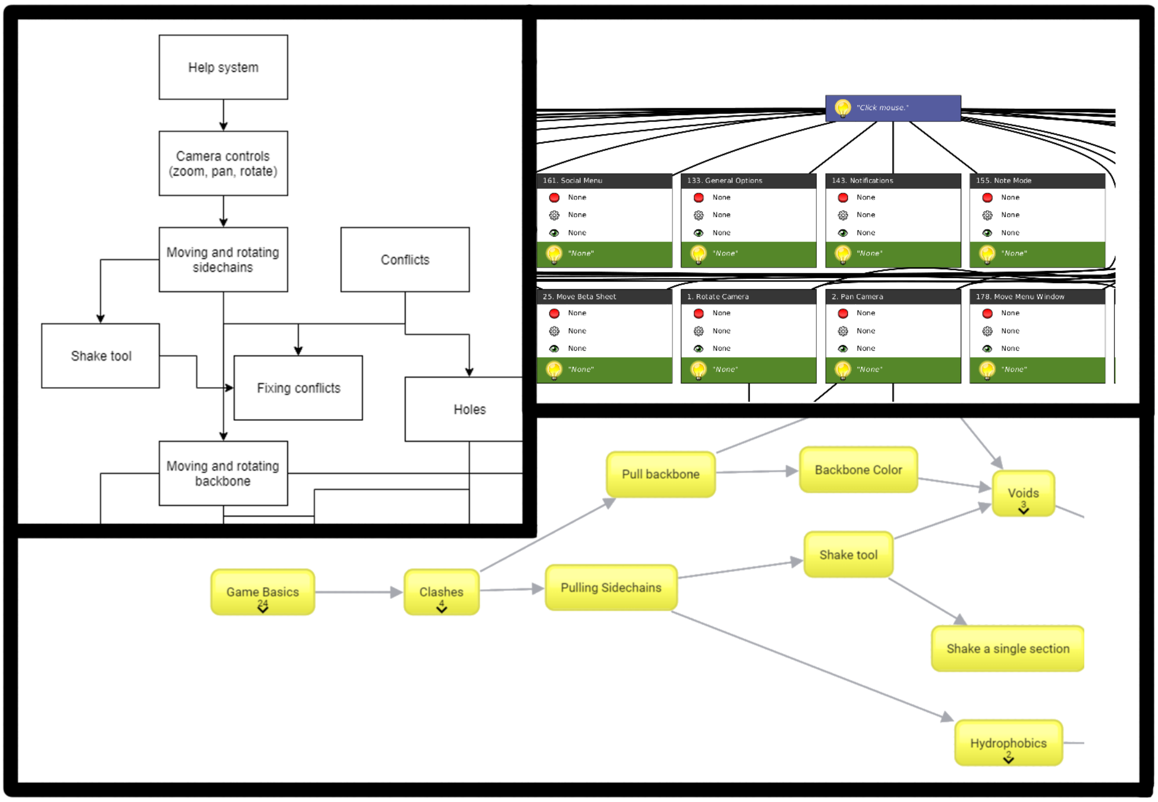


Figure 7.2: Examples of describing the process of playing at a surface level. Top left: P3 (*Foldit* Novice) diagrams the camera controls and basic mechanics of *Foldit*. Top right: D2 (*Foldit* Developer) describes basic skills and game elements such as "Click mouse," "General Options", and "Rotate Camera." Bottom: D1 (*Foldit* Developer) describes basic game elements similar to P3. Surface-level descriptions were most common in developers, novices, and *Eyewire* players, though *Foldit* and *Eterna* experts occasionally gave surface descriptions as well. Note that these figures are excerpts from skill chains created by participants.

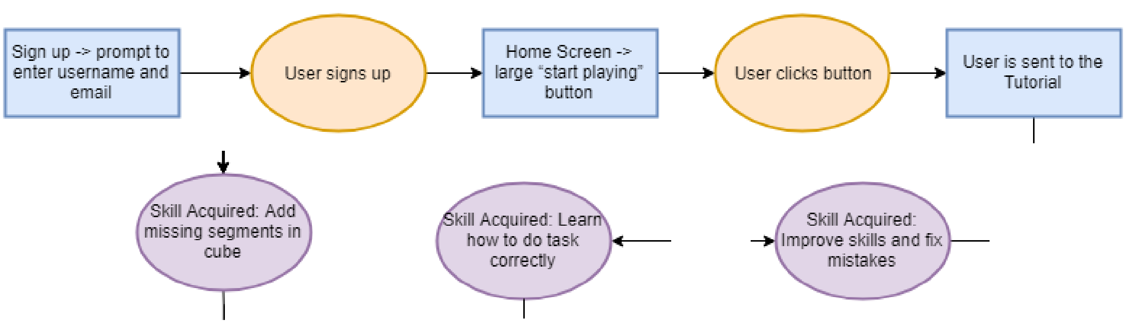


Figure 7.3: Excerpts from D3 (*Eyewire* Developer) *equating the tutorial with the onboarding experience*. Top: the beginning of the chain procedurally walks through the tutorial experience. Bottom: the developer assumes that once a skill has been introduced, the player has acquired that skill. Note that this figure is made of excerpts from a skill chain created by a participant.

assumptions that every player experiences the tutorial design as intended by the developers, assumptions that a concept has been learned (and mastered) as soon as it is introduced, and assumptions that the end of the tutorial is the end of the learning and onboarding process. This can be seen in how players and developers consider the divides of the skill chain to be based on how the tutorial is laid out, how they equate the acquisition of tools (new in-game abilities) with skill progression, how they procedurally describe the step-by-step flow through the tutorial (sometimes quoting the game directly), and how their rigorous procedural description of the skills of the game are dropped beyond the tutorial. See Figure 7.3 for an example. This pattern was less common in players with expertise, including *Eterna*'s player/developers.

The second pattern observed in this theme was participants' descriptions of *a prototypical novice's journey to expertise*. In considering the tutorials as standard experiences, participants would predict, assume, or interpret what the game expects of the player, and then describe the discoveries that a novice player "should" be making. They engage in discussion with a theoretical novice (such as through Socratic dialogue), raising thoughts from a theoretical novice's perspective and providing guidance as if writing for a novice to learn from the skill chain itself. Sometimes the skill chain would even be structured to be used as guidance for a novice.

> *"Start: What am I looking at... The Protein (I assume)... Okay, so there's point's [sic] and stuff. Gonna need to raise it... But how do I actually move things?... So this score changes in real time based on what I'm doing. Noted. I don't wanna sit here and drag every sidechain though... It'd be pretty tedious and boring if you had to go through and manually drag every sidechain, so we have the Shake tool! ... Wiggle is awesome! Why don't we just use this all the time? ... Situations where Wiggle doesn't work..."* (P10, *Foldit* Player)

Notably, because *Eterna*'s tutorial structure is different than *Foldit*'s and *Eyewire*'s, this sub-theme was expressed differently for *Eterna* chains. Instead of stepping through tutorial levels, *Eterna* chains explained that one learns the basic game concepts and mechanics, then learns the scientific underpinnings of the game and the game dynamics introduced by that, and finally participates in "lab" challenges for scientific contributions. In this way, the "tutorial" is more spread out across a player's journey to intermediate expertise, but was still mostly consistent across all chains produced by both players and developers. The passivity of the tutorial was also present, though latently expressed. Through their skill chains and interviews, the fact of learning basic concepts in the tutorials seemed to go unquestioned. On the other hand, for advanced concepts which the participants believed were separate from the game's onboarding (and which are not present in the tutorial), participants struggled to explain how they learned them, mostly citing exploratory and social learning.

7.1.4.4 Knowledge Framing

This theme captures meta-level frames around the skills that participants described in their skill chains. Specifically, participants attended to the motivations of playing and used structural markers to provide relational metadata.

Participants frequently framed the skill chain through the lens of *motivation*. They described their goals and motivations of play, of science and contribution, of socialization and social rewards, and of the game systems and game rewards, as well as describing the process of discovering these motivations.

> *"User receives points and is shown place on the leaderboard... Motivation increases"* (D3, *Eyewire* Developer)

> *"achievements... millionaire milestones..."* (P12, *Eyewire* Player)

> *"High score... Competing online... Contributing to science..."* (P3, *Foldit* Novice)

Interestingly, despite the frequency of references to motivation for *Foldit* and *Eyewire* chains, there were absolutely no references to motivation in any *Eterna* chains, both for players and developers. Yet, when member-checked via interview, players (including player/developers) expressed the same motivations for playing as *Foldit* and *Eyewire* players. We expect this omission is due to a lack of strong gamification elements in *Eterna*. Although players are motivated by the intellectual puzzles and scientific contributions, because these motivations are not linked directly to the flow of "progression" through the game (besides unlocking the "lab" challenges), the reasoning behind engagement goes unmentioned, as if tacitly understood that everyone knows why they are playing and so it does not need to be said in a diagram of how to play. However, this result could also be simply due to the task framing or small sample size. During post-hoc discussions, one developer wrote:

> *"It seemed to me that motivations were an answer to the question "Why do/did I learn (skills)", and not to "How do/did I learn". So it never occurred to me to mention this aspect of the process...."* (D4, *Eterna* Player/Developer)

The last two findings relate to the structure of the skill chain itself. Unlike traditional skill chains, in which each node represents a different skill, participants often used a *flowchart-like structure* with *structural nodes* for additional organization. The flowchart approach was used to map the decision-making process. For example, P8 (*Foldit* Player) includes nodes "Is it an ED?" and "Is it a dimer, trimer, etc?" Based on these nodes, we believe these participants were attempting to create a decision tree rather than a skill chain. This is perhaps because a decision-based approach directly follows their line of thinking, whereas a skill chain requires higher-level analysis on their part. Notably, though, there were no clear circular skill dependencies[8] (as flowcharts and decision trees sometimes have) which might have suggested the need for repeated practice or dovetailed task variation; instead, all chains with nodes[9] were directional acyclic graphs, though they sometimes had multiple start and end nodes. Structural nodes, such as "Advanced techniques" (D1, *Foldit* Developer), "More successes" (D4, *Eterna* Player/Developer), and "Main Techniques (Hand-Folding)" (P10, *Foldit* Player), were used to organize the hierarchical structure of information and to show relations between concepts. Sometimes titles were given to sections of the chain, such as "Tutorial Stage," (P16b, *Eterna* Player) and "*Foldit* Design" (P10, *Foldit* player). As discussed in Section 7.1.5.1, these results reveal that skill chains may yet be ill-defined. Despite methodological limitations about how we prompted participants (also discussed below), our study calls into question what it means for something to be a skill.

7.1.5 Discussion

This study examined the methodology of directly eliciting skill chains from players and developers for ECCSGs. In the section below, we address our two research questions, the potential implications for design, and limitations of the study.

7.1.5.1 RQ1: How do players and developers conceptualize the skills gained through play?

Four models of skill chains were observed, though most chains used a combination of models. The first, used by P2, P3, P13, D1, D2, D4, and D6 could be called **tutorial-oriented**, as it lays out the elements based on how they are introduced in the tutorial. The second, used by P1, P5, P6, P8, P9, P11, P12, P14, and P16a could be called **core loop**, as it focuses on only what's involved

[8]There were two instances of circular flow, both describing gameplay procedures, such as "Shake" and "Wiggle" pointing to each other, suggesting one may alternate between them during play. However, these instances did not seem to treat the nodes as skills, instead treating them as steps in a protocol, hence we conclude they were attempting to document a single procedure.

[9]Other submissions included a plaintext list and a spreadsheet.

in the core gameplay loop. This agrees with the finding from Horn et al. [218] that "skill chains run together in a core mechanic." What Horn means, as he explains later, is that the skill chain tapers and culminates in an overarching goal, which we also saw, as many player chains ended in something along the lines of *get a high score*. However, it's also true that chains in this model focus on the central mechanics of the game. In the case of *Foldit*, for example, chains are built around usage of the Shake and Wiggle tools for resolving the most typical problems of the game's puzzles. In this way, several of the participants' chains resembled flowcharts or decision trees more than skill chains, since the core loop is based on a web of decisions and actions rather than hierarchical skill requirements.

The third model, used by P7 and P8 could be called **stream of thought**, as they include disconnected tips and discoveries that seem streamed from the player's consciousness. The last, used by P10 and D2 could be called **WYSIATI**, or What You See Is All There Is [248], as these skill chains attempt to include *every* visible game element, trying to categorize them into a larger structure. These chains resembled concept maps in how they attempted to draw connections between all game elements and surface-level concepts.

Across these four models of skill chains, 18 conceptual components (sub-themes) were observed. Skill chains included knowledge structuring through *flowchart structure* and *structural nodes*. They emphasized the *motivations* for engaging with the game in the first place and the *big picture* of understanding the context of play. The bulk of the chains consisted of the *surface process* and *detailed process* of playing. Sometimes, participants would *equate the tutorial with onboarding* or frame the skill chain as *the prototypical novice's journey to expertise*. Players highlighted *social learning and socialization* as a key component of onboarding, especially through references to *community-created knowledge* and the *use of paratexts*. They demonstrated *self-reflection* on how *their observations led to their current behavior* through the *discoveries they made* and *background knowledge* they learned while *gaining an "eye"* for the nuanced mental models that went into their decision-making and strategizing. This process required *dedicated practice* and led to *applying situational strategies*, drawing from their wealth of experience on what tools and strategies are effective in each kind of situation, and what the problem space is.

Ultimately, these results highlight that players and developers typically don't see the sets of dependencies between skills. Rather they see the process of progressing and playing from an experiential perspective.

Interestingly, unlike Horn et al. [218], we did not find that skill chains remained flat (i.e., broad branching dependencies without much depth in the chain). However, the games examined in this study were more complex than *Paradox*, the game they studied. Moreover, our goal was not to produce a single, comprehensive skill chain that encompasses all elements. Arguably, attempting to include every element of the game into a skill chain (P10, D2) will result in a mostly flat chain due to the phenomenon that critical, deep factors are inherently sparse (cf. Zipf's law, the power law, or the Pareto Principle [4]). Because of this, complex skill chains will be rare; more commonly, a game element will have little depth beyond a surface-level description of its purpose.

In comparison to the recent work by Hesketh and Deterding [213], our findings relate closely to theirs. Both studies found that expertise involves the use of paratexts, exploratory learning, practicing in different game modes (such as *Eterna*'s puzzle maker), using add-ons (such as the scripts in *Foldit* and *Eterna*), learning from community content, mastering the basic controls and mechanics, and learning/applying non-game-specific knowledge (in this case, scientific background knowledge). These results also agree with the classic case study of Apolyton University, the player-made learning hub that demonstrated social learning, cognitive apprenticeship, and knowledge organization within the context of video game expertise [486]. Social learning has also been previously identified within *Foldit* by Bauer and Popović [35]. Through post-hoc analyses, they show a correlation between collaboration (i.e., joining a group) and improved personal performance, as well as a correlation between early collaboration and increased participation.

Overlapping Conceptualizations Between Players and Developers One interesting sub-question of RQ1 is: to what extent do players and developers overlap in their definitions of a skill chain? We found a large amount of conceptual overlap in skill descriptions between players and developers; however, the overlap reflects only the way in which the existing tutorial describes the skills, with no confirmation that the tutorial's approach captures an underlying truth. The developer chains mostly *equated the tutorial with onboarding* by procedurally describing the *surface-level process of playing* and going no further than the end of the tutorial. Several chains also expressed an assumption of what could be called "once-and-done learning," in which a skill demonstrated once is assumed to be fully mastered. Similarly in prior work, skill chain developers have used single behavioral instances of demonstrating a skill to assume that the skill is acquired [251], though other more player-centric work represents this more gradually [26].

Novice descriptions were similarly at a surface level, which marks a curious connection: developer chains were more similar to that of novice players than of expert players, with the exception of *Eterna*'s player/developers, who were more similar to other experts. Perhaps this is because the tutorial is designed to reflect how the developer understands the skill chain and the novice understanding reflects the tutorial. Both developers and novices quoted the game's instructions verbatim, and while novices considered mostly surface elements, developer skill chains were entirely focused on concrete game elements and interactions. Developers described the game concepts, visuals, and elements to learn, seemingly concerned only with the core mechanics rather than the nuanced dynamics that emerge, or the nuanced mechanics that play into the core. This can be seen in how D3 (*Eyewire* Developer) procedurally describes the tutorial screen-for-screen, or how D1 (*Foldit* Developer) chunks important details into "Game Basics" and "Camera," whereas *Foldit* expert players (P9 and P10) unpack these mechanics in far greater detail.

Eterna player/developers, on the other hand, take time to describe these details, suggesting that this is not a trait of all developers, but rather of developers without deep expertise at their own game. One limitation of this observation is that no novice *Eterna* players were present in this study for comparison. However, our claim here is agnostic to both the participant's status as a developer and which game they are playing: experts provide deeper descriptions and novices provide surface descriptions. The fact that only *Eterna* player/developers demonstrate this distinction is an artifact by the nature that they are the only developers who are also experts in our study.

The surface-level descriptions from novice players are an intuitive finding. The first steps of learning how to interact are the basic controls: nearly all spatial games begin with controlling movement. Novice players don't possess the mental models to elaborate on the game beyond this (P2, *Foldit* Novice). This finding agrees with two results of prior work [218]: that novices are quicker to identify low-level interface and gameplay skills, and that skill chain analyses surface low-level and pre-existing skills, for example, in descriptions of the controls and fundamental background concepts that contextualize the gameplay experience, such as motivations for playing and task overviews.

Expert players, on the other hand, give some attention to the early concepts because they are pervasive and/or explained often, but they also attend to intermediate concepts that are practiced often, elaborating on the mental steps to understanding tool usage (P8, P9, P10).

The Ill-Definition of Skill Chains As alluded to earlier, this work calls into question what the definition of a skill is in the context of skill chains. Although our prompts for the participants were open-ended, thus removing the guarantee that we would receive "valid" skill chains, we received evidence that complicates Cook's original definition. Namely, according to Cook, a skill chain is definitionally a hierarchy of skill atoms, each atom containing four components: a player action, a game simulation, the game's feedback about the updated game state, and the player's internal mental model update [95]. By Cook's model, skill chains are meant to capture the entirety of player learning and interaction for "pretty much any game imaginable" [95, p. 4]. Yet, how can we capture strategies in this model? Or decisions? These, too, are skills that the player needs by the common definition of skill.

The categories generated in Table 7.2 are one potential avenue of expansion. By creating new node types, such as the distinctions between Actions, Procedures, and Strategies, we may be able to construct more meaningful, nuanced hierarchies of player learning. However, we note that these categories were generated to explain latent intent of our participants rather than for direct use in traditional skill chains themselves. For this reason, our categories have conceptual bleed between skill chain content and the larger cognitive and social contexts of acquiring and sharing expertise. Guidance and Discoveries, for example, are markers of the participants' own conceptualizations of their learning, not skills that can be tracked.

Yet these seemingly ancillary categories represent critical relational metadata needed to adequately explain how skills build on one another. Without the background information on how participants received Guidance or made Discoveries, skill chains are missing a fundamental context to make sense of the skills themselves. For this reason, researchers building on this work may be interested to turn to instructional design models such as Four-Component Instructional Design (4C/ID) which offer this kind of nuance in, for example, how 4C/ID uses skill decomposition to break a complex task into constituent skills and scaffold training with supporting information [540, 537].

7.1.5.2 RQ2: How effective is free recall as a method for directly eliciting the skill chain of a ECCSG from players and developers?

Contrary to our hypothesis, free recall seems no more suited to this (more structured) context than other cognitive task analyses. The skill chains elicited either reflected the existing tutorial (which does not inherently capture the skills needed to play) or captured errant thoughts from expert players that do not sum to a coherent hierarchy. Horn et al. [218] similarly found through their method that skill dependencies were unclear and confounded by level design.

The creation of accurate and thorough skill chains remains a difficult process. However, we argue that this method has value beyond skill chain elicitation. Rather than being used for generating skill chains, explicitly asking the players about their expertise provides a window to the forefront of their minds: what their common core loop is and their most recent gameplay experiences.

Further, direct elicitation provides evidence that although expert players are unable to retrieve a coherent compilation of their knowledge without prompting, they can retrieve some of the salient points. Some of the most interesting nodes in expert player chains were disconnected from the rest of the chain with no connecting edges — yet those nodes were written down as something on the expert's mind (P7, P9, P12).

And, as described in Section 7.1.4.2, the descriptions elicited by free recall can be used for instructing new players, summarizing the tutorial, onboarding new or external developers, and understanding the players' expression and structure of core gameplay information.

It is worth noting here that our method of direct elicitation may have been more suited to elicit decision trees than skill chains. Different CTA techniques can produce a variety of outputs, and decision trees (as well as related outputs like process diagrams) are one of the easier forms to elicit, second only to concept maps [216, 420, 103], which we also saw from the dataset, especially the *WYSIATI* chains.

Although this study was not effective at eliciting the expert skills themselves, it was effective at understanding how players conceptualize skills. For example, P10 (*Foldit* Player) refers to some skills as *tactics*. Earlier during the multi-coder analysis, tactics were considered to be situated with procedures. Recall also that in related work, tactics were situated as a short-term composition of actions, of which strategies were composed at the highest level [30]. Moreover, Horn et al. [218] found that the distinction between procedural and strategy skills was fuzzy. However, based on the player's description, there appears to be four levels of action-/decision-making:

1. **Actions** are the lowest level of interaction, mapping to a single input or atomic interaction. Actions have binary success and require minimal physical effort to execute.

2. **Procedures** are sequences of actions routinely strung together in a particular order or combination, such as combos in fighting games or complex maneuvers in platformer games like triple jumps and wall jumps. Procedures have binary success and can require dexterity to execute properly.

3. **Tactics** are procedures, often longer or more complicated, that are *open to interpretation*, such as inputting a combination of actions in which the optimal order is ambiguous, or adjusting the parameters of an action (such as duration of a button press) resulting in a gradient of success in the larger context of the tactic, or even choosing which procedure to execute. Rather than

having binary success, tactics can range in effectiveness and can require both dexterity and cognition to execute.

4. **Strategies** are high-level plans and decisions which inform the tactics used. Like tactics, strategies can range in effectiveness, though they typically require only cognition to plan and execute. The dexterity involved comes from the tactics which compose the strategy.

Yet, these descriptors remain ambiguous, and participants likely struggled with this ambiguity as well while performing the task. Much of the previous work on skill atoms (e.g., [219, 448, 26, 134, 520]) considers only physical-, dexterity-, or declarative knowledge-based skills with outward action (e.g., pressing a button or inputting the correct answer) but do not describe decision-making in detail. Is the decision to choose between skills itself a skill? That is, consider the possibility space where, on one end, choices are strategically unique and have an objective value ordering (thus having an optimal answer), and on the other end they are strategically identical, differing only in aesthetics (thus being entirely preference-based): at what point along this spectrum does the decision change from the player's preference among similar outcomes to there existing a unique, correct answer? Once there exists a correct (or even "more correct") choice, the player's decision — we argue — is skill-based and ought to be captured in skill models. Yet, decision-making often falls within the realm of non-routine problem-solving, while most game skills are routine [540, 172, 170]. This ambiguity is a shortcoming of the current definition of skill chains, and future work can disambiguate this further.

In the categories developed through the multi-coder codebook thematic analysis, tactics were ultimately grouped with strategies. However, given how P10 emphasized tactics by name, it is worth considering tactics as a separate type of skill between strategies and procedures.

Direct elicitation as a method was also helpful for understanding the use of jargon in the context of game expertise. Jargon was mentioned in three ways: alluding to the language of the game without defining it (e.g., P5, P11), learning and explaining the jargon (e.g., P9), or making up jargon to refer to game concepts in a language they understand (e.g., P3, *Foldit* Novice, calls clashes "conflicts" and voids "holes"). Although there is not enough evidence to make claims about this language use, further exploration may lead to a better understanding of the cultural assimilation of novices with respect to learning and using the language of the game.

7.1.5.3 Reflections on Methodology

Through member-checking interviews, we discovered that participants generally spent between one and a few hours on their skill chain, sometimes across several days. Often, they replayed the tutorial beforehand to refresh their memory on how the core concepts were introduced, which adds context to the theme *Tutorials as Passive and Standard*. During interviews, participants described many more skills than they listed on their chain. When asked why they didn't include these details, the same general response was given every time: doing so would increase the complexity of the diagram exponentially. (Understandably, no participant expected to need to spend so much time diagramming every minor detail of the game they've played for several years.)

Another trend that emerged through member-checking interviews was an uncertainty about how to draw the skill chain in the first place. As one developer described:

> *"First, I'm not sure how to go about vocabulary. I used the word "beads" in the beginning, because that's how they felt to me as I started playing, when I had absolutely no idea of biochemistry and/or thermodynamics. Later, as I understood better what they were supposed to model, I started calling them "nucleotides" or simply "bases". So the question would be: what vocabulary should we use in this document? "total noob" or "accurately scientific"? Beside that, I have stopped at the stage where players can solve challenging puzzles, but the game goes on with increased difficulties, in particular, multi-state puzzles. Is it meaningful to talk about that in this document? Finally, there's the whole 'labs' domain of the enterprise, but this is no longer a game I think, and participants have vastly different approaches and experiences with it..."* (D4, *Eterna* Player/Developer)

This self-reflection on the methodology highlights its challenges. Without guidance from a CTA expert, it is unclear what vocabulary to use or how much detail to add to the skill chain, especially for the latter scientific aspects of the game. Thus, if one intended to use this method practically, participants would require guidance on these uncertainties, as CTA protocols often note [103].

7.1.5.4 Preliminary Toolkit for Skill-Based CSG Design

Based on this work, we synthesized the categories and themes created into six potential design suggestions for CSG developers to support learning. This section comes with several caveats. First, these takeaways are not empirically tested; rather, they represent our own practical interpretations of the themes generated, which we derived by identifying the ways in which players develop and articulate expertise and considering what design patterns would promote the observed, existing learning processes. Second, because we are deriving these implications from a latent analysis of players' own self-reflections, this section is focused only on paths to expertise and not skills themselves; these recommendations should be read as distinct from skill knowledge and the use of skill chains. Lastly, these takeaways may be limited in generalizability, as it is unclear to what extent these learning processes extend beyond the CSGs studied.

That being said, from the sub-themes of *Experts are Experiential Learners we recommend CSG developers* **give the big picture up front** to set the groundwork for contextualizing the rest of the game. This corresponds with van Merriënboer's 4C/ID model, which puts the focus of learning on completing whole learning tasks (i.e., the big picture) from the very beginning of the learning process [540].

Second, **embrace social learning and paratext use**, such as by adding features to support player dialogue and integrating external resources into the game proper. Not only is collaboration a critical incentive to playing CSGs [114, 229, 241], it supports skill practice through peer modeling and cognitive elaboration [284]. Moreover, both early success and early collaboration have been shown to correlate with increased participation [35]. Although the causality of this effect is yet unclear (perhaps more skilled or extroverted players are preinclined to participate), it may be beneficial to start players with a positive and social experience.

Third, **reinforce the intended structure of knowledge**, such as through visualization of the hierarchy of concepts (cf. skill trees in roleplaying games like the *Elder Scrolls* series [483], which are used to introduce concepts over time and visualize hierarchical progress). Clarifying the relations between concepts can help avoid "horizontalization," whereby each fact or concept is given equal and sequential attention, which is often disadvantageous to constructing a proper mental model [284].

Fourth, **situate learning within applicable, meaningful contexts**, since expert strategies are most often situational. This can be achieved through tasks designed to test the player's knowledge of a particular concept or ability to execute a particular tactic, supported with just-in-time (JIT) information [177, 537].

Fifth, **design for discovery and self-reflection**. Discoveries trigger the generation effect [479], which promotes retention, and self-reflection promotes integration [537, 540, 314]. One way to implement this in design is to teach through active learning or systems exploration, wherein the player engages in an observe-experiment-evaluate cycle, as opposed to being told by the game what to do [562].

Finally, **encourage practice and learning beyond the tutorial**, since novices and developers seem to *equate the tutorial with onboarding*. This can be achieved through blended tutorials — as seen, for example, in *Portal* [535] — which blurs the line between where the "tutorial" stops and the "game" begins. An alternative, complementary approach is *designed challenges* and practice spaces, such as chess problems ('compositions'), sandbox modes, and play vs. AI. These methods provide supplementary ways to hone skills beyond the entry-level tutorial and encourage learning, especially in combination with social features as described above. Note that *Eterna*'s use of the latter strategy, with its puzzle maker feature, is praised by expert players as a major contributor to their expertise.

7.1.5.5 Limitations and Future Work

First, this work has limits of generalizability. We examined only three games and thus may not generalize beyond the small niche of ECCSGs. Moreover, we considered only four novice chains, which may not have been enough to reach theoretical saturation. On one hand, in Horn et al. [218], the authors note that skill chain elicitation is quickly saturated, requiring only five participants for them to reach saturation, so these sample sizes may not be too far from saturation. On the other hand, the method of Horn et al. was more directed, so an open-ended elicitation method may need a larger sample to reach the same level of detail. In addition to testing larger samples, future work may consider a crowdsourced version of this elicitation method, i.e., allowing a crowd of players to collaboratively build a single skill chain, perhaps with guidance by the CTA practitioners to develop a consistent vocabulary.

Second, the results produced may be affected by a lack of instruction for the participants. Much like unaided free recall, participants expressed ambiguity in what output was desired, which led them to make assumptions about the desired format and content. This was, of course, desirable for the purpose of an exploratory method, but this ambiguity may affect reliability and reproducibility. In the case of developers, for example, they described their chains as focusing on game mechanics and their intentions for the tutorial, rather than other game aspects (such as social components) or the current tutorial experience. This ambiguity led to our outputs having a variety of formats, such as flowcharts, decision trees, process diagrams, and concept maps. The diversity of outputs may therefore be considered an artifact (as opposed to a finding) of free recall which is known to result in an incomplete representation of tacit knowledge [86, 103]. However, as stated earlier, free recall was chosen because the present context is more structured and therefore was hypothesized to have been more suitable to free recall, though it was not. Future applications of direct skill chain elicitation may consider specifying what is or is not desired as output, with more examples than were provided in this study.

Third, the qualitative analysis is subject to bias, especially because the primary coder is also a ECCSG developer. The coding was performed as impartially as possible but with this bias in mind. It is for this reason that we provide the dataset and coding audit trail so that our scientific peers may check and validate this work.

Lastly, the results may also be biased by the effectiveness of the current tutorials. While these results suggest strong prevalence for social and exploratory learning, especially through the use of paratexts and trial-and-error, these learning patterns may also be simply indicating a failure of the current tutorials to provide other means of onboarding. Therefore, future work should examine whether players are learning in these styles because social/exploratory learning is inherently effective or because the instructional designs of the tutorials were extremely ineffective, causing other approaches to be favorable by comparison.

7.1.6 Conclusion

This work attempted to directly elicit skill chains of CSGs from players and developers via free recall in order to understand how they conceptualize the skills and skill dependencies of the game. We identified nine types of skill chain nodes: Actions, Practice, Procedures, Strategies, Guidance, Discoveries, Social, Objects, and Motivation. Four major themes were found in participants' skill chains: the process of gaining expertise as *experts are experiential learners*, an emphasis on *the process of playing*, a conceptualization of *the tutorial as passive and standard*, and insights into the *knowledge framing* around the skill chains. We conclude that players and developers overlap partially in how they conceptualize skill chains, both with each other and with existing skill chain models. Although free recall was found to be ineffective for determining a traditional skill chain, it still produced implications for ECCSG skill-learning design based on player and developer conceptualizations and was able to elicit the core gameplay loops, tutorial overviews, and some expert insights.

7.2 A Skill-Based Cognitive Task Analysis of Foldit

From the previous study, it became clear that I would need to actively pursue an understanding of *Foldit*'s skills in order to create a more comprehensive skill chain. Furthermore, in addition to other various insights into how players view their skills, a key takeaway was that I needed to frame — from very early on — how these skills fit together.

Therefore, I conducted a Skill-Based Cognitive Task Analysis (SBCTA) based on the framework by Seamster et al. [467] to comprehensively model *Foldit*'s skills. This process was grounded in the instructional design theory of 4C/ID because this theory has demonstrated success in modeling complex learning over the course of decades of research [537, 536, 542, 539, 540]. Although 4C/ID is more often used in medical and military fields, the prior work of this dissertation has shown that playing ECCSGs is incredibly difficult — perhaps so difficult as to be comparable to surgery or rocket science; so let us use the techniques which are designed teach surgery and rocket science.

7.2.1 Four-Component Instructional Design

The Four-Component Instructional Design (4C/ID) model is an instructional design methodology for complex learning, i.e., learning complicated, often professional, skills [541]. In more recent years, van Merriënboer has presented this model as a systematic ten-step guide for designing a complex learning curriculum [540]. This positions the 4C/ID model at the high level of course design [539], as opposed to lower-level theories which provide guidelines for instructional message design such as Mayer's cognitive theory of multimedia learning [317] and Sweller's Cognitive Load Theory (CLT) [500], or the even lower-level psychological theories that describe memory systems and cognitive processes, such as Paivio's dual coding theory [82] and Baddeley's working memory model [28].

> *"The basic claim of 4C/ID is that all environments for complex learning can be described in terms of four interrelated components: (1) learning tasks, (2) supportive information, (3) procedural information, and (4) part-task practice. Learning tasks are meaningful whole tasks, based on real-life tasks from professional or daily life and typically require the integrated use of knowledge, skills, and attitudes. Supportive information helps learners to perform the problem-solving and reasoning aspects of these tasks. Procedural information points out to learners how to perform the routine aspects of such tasks. Part-task practice is additional practice to develop routine aspects of the tasks to a very high level of automaticity."* [542]

There are many smaller claims within 4C/ID that shape implementation, including (but not limited to) simple-to-complex ordering, fading-guidance strategies, promoting compilative processing (chunking), promoting inductive processing, rule-based instruction that highlights common structural features, problems with non-specific goals (ill-structured problems), JIT (just-in-time) information, annotated worked examples, partitioning, demonstration, scaffolding, elaborative encoding, and so forth [541, 539, 542, 537]. These principles overlap strongly with heuristics of playability and tutorial design, as they should if they are drawing on the same fundamental psychological phenomena. Though the details are beyond the scope of this review, the current design science research draws on these principles, among others: the physical-fidelity principle, the training-wheels principle, the variability principle, the collaboration principle, the completion-strategy principle, the prior knowledge activation principle, the multimedia principle, the dynamic visualizations principle, the redundancy principle, the coherence principle, the self-explanation principle, the self-pacing principle, the modality principle, the temporal and spatial split-attention principles, the signaling principle, the segmentation principle, the component-fluency principle, the individualization principle, the second-order scaffolding principle, and the development portfolio principle [539, 540].

7.2.2 Methods

Foldit expert players (n=12) were recruited via *Foldit*'s chat channels and direct messages. Participants were offered a $15 USD Amazon gift card as remuneration. Participants were asked either to attend a focus group or record a gameplay experience and attend an interview about their recording. In total, the dataset included: one focus group of four participants, three individual interviews, one interview of four participants who wanted to participate as a team, and five recordings. All methods were approved by Northeastern University's Institutional Review Board.

7.2.2.1 Focus Group

Four participants attended a virtual focus group for a diagramming activity. During this meeting, the participants and I collectively viewed a virtual collaborative whiteboard on Miro.[10] Participants were asked to diagram the progression of skills that they use in *Foldit* and were given a preliminary diagram which I designed as a starting point. Participants edited the diagram and added their own ideas while I moderated the activity. See Figure 7.4 for the completed diagram.

7.2.2.2 Interviews

Prior to interviews, I asked participants to share a recording with me of their gameplay for a specific task. The exact task depended on the participant's self-described expertise and the aspects of *Foldit*'s skills which I had not yet fully explored. I collected five recordings in this way,

[10] `https://miro.com/`

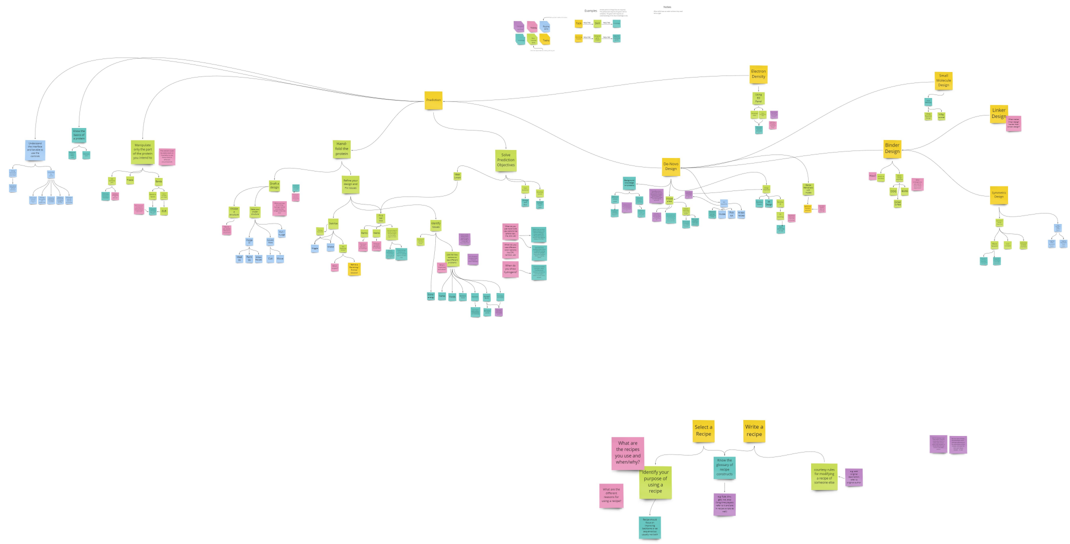


Figure 7.4: The final skill diagram on Miro. This screenshot, captured at an illegibly small distance, shows the extent of detail participants provided in diagramming the skills they understand for play.

one for each interview (only one participant submitted a recording for the group interview) and one additional recording from a player who declined to be interviewed.

During the interviews, I asked the participant to describe the intentions behind their behavior, for example why they took a specific action, what they were thinking about at a particular moment, or what they were looking for when they were adjusting the camera or visualization settings. Through these interviews, I developed a "gold standard" protocol [87] which was reviewed with the participants and developed iteratively across them (e.g., by asking participants if they agree with the procedures as described by prior participants). In the final (group) interview, I collected feedback from the participants on both (1) the skills diagram from the focus group, and (2) a design document outlining my intentions for a revised tutorial progression. In this way, I confirmed with expert players that my tutorial re-design aligned with their expectations for how *Foldit* requires various skills from its players.

7.2.3 Results and Discussion

The output of this study is not something easily visible or generalizable. The intention was to produce an understanding of skill-based learning specific to *Foldit*, and this goal was achieved.

Main Procedure: Prediction Task

1. Prepare the fold
2. Draft the secondary structure
3. Refine the fold

Subprocedure: Prepare the fold

1. Set view options:
   — Turn on sheet bonds
2. IF secondary structure has NOT been provided by the puzzle:
   — Set View Option: Color to Hydro
   — Identify alternating (blue, orange, blue, orange...) sidechains;
   assign them to sheets
   — Identify patterns of repeating sequences of 7 consisting of
   no more than 2 blues or oranges in a row; assign them to helices
   * To check correctness, when idealized as a helix,
   one side should be blue and one side should be orange
3. Using selection mode, for each helix and strand:
   — Select it and use Ideal SS (default hotkey 5)
4. For each loop:
   — Make cutpoints at the edges of loops in order to set them aside temporarily

Figure 7.5: Sample of the gold standard procedure generated by the SBCTA.

Curious readers can view the gold standard procedure I developed at `https://osf.io/rmext/` or refer to Figure 7.5 for a sample of the output, though most of this documentation relies on expert *Foldit* jargon.

Similarly, through this iterative design process, I produced a preliminary tutorial design, available at the same link. (See Figure 7.6 for a sample of that output.) The value of this study, then, was that I created a validated design plan in preparation for implementation in the next chapter.

The tutorial design process made use of the Tandem Transformational Game Design framework [516, 112]. This framework was designed for the development of serious games and consists of multiple loops of iteration. In the game-driven goal delineation loop, one iterates between delineating the goal (here, deciding what to teach and through what methods) and conduct-

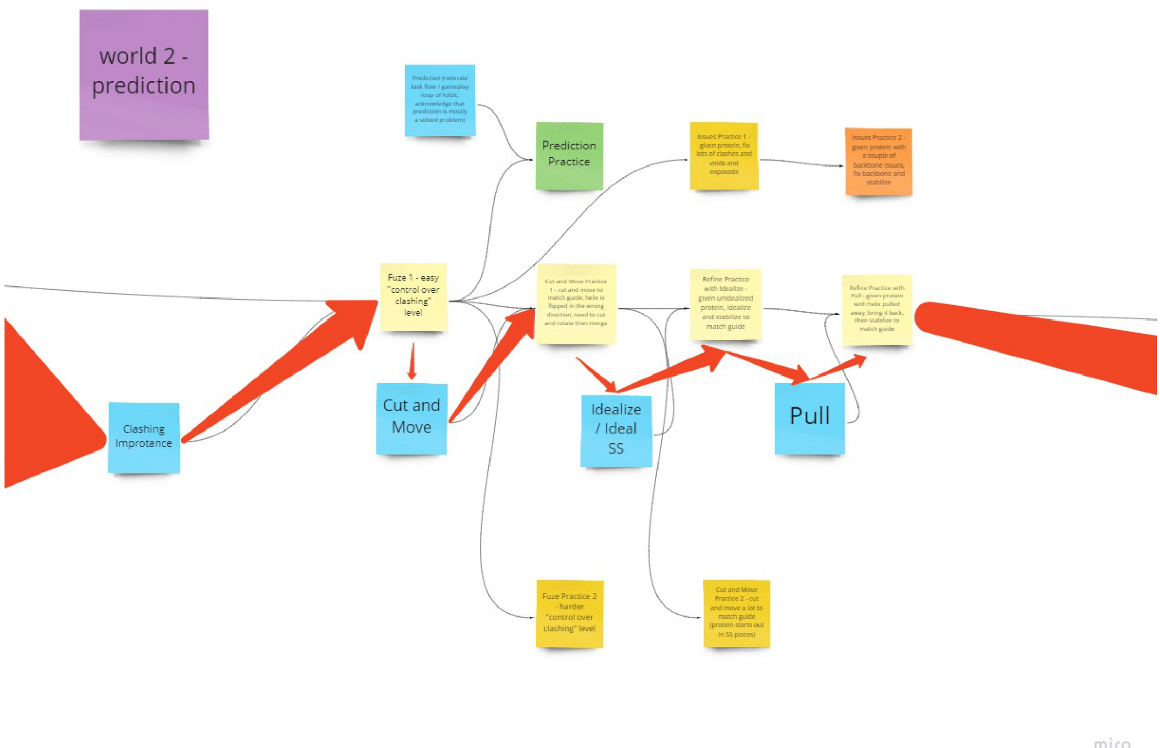


Figure 7.6: Excerpt of the tutorial design plan. This image shows level set ("world") 2 of 15. Tutorial levels are in blue; primary practice levels are in yellow; secondary (optional, more difficult) practice levels are in darker yellow and orange; science levels are in blue. The thin black arrows show prerequisites (i.e., a level is unlocked when all of its prerequisites are completed); the thick red arrows show the recommended (by the game system) path through the levels.

ing literature reviews. In the goal-driven game design loop, one iterates between prototyping and playtesting. Between these loops, one iterates between bringing them into alignment and separately working on each process. In this way, I iterated between collecting data from participants, reviewing research suggestions (both from my prior work and other literature), designing a tutorial plan, and validating that plan with participants.

While this work is intentionally not generalizable, I can reflect on methodological insights: how difficult was it to apply SBCTA, and how valuable were the outputs? Should other ECCSG developers take this approach?

From my experience, this method is very niche. It can provide very detailed, very specific workflows from expert players which may not always be in agreement with each other. This method may be more useful when a researcher or developer is interested in an individual's workflow or to see which mechanical details players fixate on. For me (after years of researching *Foldit*), there was little said by my participants which was wholly surprising to me, but in following SBCTA protocols, I was able to elicit specific details on the microscopic choices they make during play.

In the focus group, participants struggled to elaborate on details without more specific prompting. For example, one wrote, "turn off [view] options to reduce noise [and] focus on what[']s important," but did not elaborate on what is "important." Instead, participants sometimes added comments directed to (or written for) developers — they described what tools or features would help them with their workflows. Their behavior here echoes the findings of Jagex [372] who identified that input from expert players is limited — because they are deeply invested in the game, they often do not imagine it beyond what it already is, and instead focus on quality-of-life details for the existing workflows (see also [333] in Chapter 4).

Should other ECCSG developers apply SBCTA? In my opinion, no. As will be described more in the next chapter, the initial tutorial design produced by SBCTA was incredibly complex: it had a total of 157 levels across 15 sets. Recall in Chapter 6, I and my co-authors recommended that CSGs simplify their mechanics and, by logical extension, their tutorials. The design produced by SBCTA was over-engineereed. It succeeded in eliciting very precise, very detailed means of teaching new players exactly what veterans understand, but because of this focus on expert knowledge, this method could not conceptualize a more succinct way to onboard players. This makes sense; in fields where CTA and 4C/ID are common (military, medical), the skills themselves are fixed, and one necessarily must teach students how the experts understand the material. Yet, for CSGs, which are capable of change and simplification, the context affords taking a simpler route. As described in the next chapter, I ultimately simplified the tutorial design. SBCTA may still have some niche use cases, such as helping veteran players share workflow ideas for very advanced problem-solving; yet, for our purpose of onboarding design, this method was too focused on expert idiosyncrasies.

7.3 Conclusion

In this chapter, I conducted two studies to deeply examine expert skills in ECCSGs, especially *Foldit* as my targeted case study. In the first study, I attempted to directly elicit skill knowledge from players and developers; this work produced several insights into how players and developers conceptualize skills and generated four themes: *experts are experiential learners*, participants emphasized *the process of playing*, they conceptualized *the tutorial as passive and standard*, and provided insights into the *knowledge framing* around the skill chains.

In the second study, I more thoroughly elicited the expert skills of *Foldit* using a Skill-Based Cognitive Task Analysis (SBCTA). Although I succeeded in eliciting expert knowledge, the output was too idiosyncratic to form a succinct and approachable tutorial design.

7.3.1 Takeaways

The goal of this chapter was to determine if player and developer conceptualizations of skill chains could help design a new onboarding experience — i.e., by using their mental models as a starting point. To this, I elicited skill knowledge from players and developers. I was able to elicit insights about the core gameplay loop, overviews of how the tutorial is already structured, and incredibly detailed expert insights into players' idiosyncratic workflows. But those insights don't provide immediate value to designing a new onboarding experience; trying to teach new players how experts think would create a tutorial that is over-engineered and overly complicated.

While I succeeded in eliciting expert skill knowledge, the tutorial design plan I created based on those expert skills was not suitable for new player onboarding. So, in the final study, I iterate on this tutorial design and try to implement as many insights from across this dissertation as possible. In parallel to the naive approach of Chapter 3, I revisit the initial goal: make *Foldit*'s tutorial more successful in its goals of teaching and motivating new players. This last empirical chapter is a culmination of the work so far and aims to answer two questions:

1. How feasible and effective are the insights of this dissertation? How do their costs and benefits compare?

2. Using Tandem Transformational Game Design, what does the process look like of improving an existing tutorial design? How can other CSG developers make use of this process to improve their tutorials?

In a sense, the following capstone study is the empirical test of all of the insights learned over this dissertation: to what extent are these design suggestions and theoretical insights actually, practically, valuable?

Chapter 8

Complex Solutions for Complex Learning

In Chapter 3, I began my investigation of identifying problems in onboarding design for ECCSGs by implementing a naive approach: can we simply directly apply theories of learning and motivation to make a better *Foldit* tutorial? The answer was, resoundingly, no. And so, I set out, across the subsequent four chapters, to identify more thoroughly what the problems in onboarding design were and what solutions might look like.

Now I revisit the original problem — making a more effective *Foldit* tutorial — armed with the insights from a few years of research. Of all of these insights, which ones are feasible to implement? Which ones are effective and efficient for the cost it takes to implement them? These are the research questions of this last study which uses the Tandem Transformational Game Design framework [516, 112] as described in the last chapter. By iteratively developing a new tutorial and documenting my process, I aim to (a) provide insights into the process of improving an existing tutorial design and (b) determine if and how other CSG developers can make use of this process to improve the onboarding of their own games.

As summarized at the end of the last chapter, this study is the empirical test of all of the insights learned over this dissertation: to what extent are these design suggestions and theoretical insights actually, practically, valuable?

8.1 Background

At this point, readers who have followed this journey will be aware of all of the necessary background information, but it is worth noting which information will be relevant here. First, this study employs Tandem Transformational Game Design and 4C/ID; see Chapter 7 for definitions of each of these. Other literature may be referenced as it becomes relevant, such as CLT and SDT (originally from Chapter 3) and various insights from Chapters 4–7.

Lastly, my usage of *Foldit*: as described in Chapter 1, I have chosen *Foldit* as a case study because it exemplifies the properties of ECCSGs which I am interested in studying. That is, Foldit is the largest and most active ECCSG, it has previously been studied for its tutorial in academic literature, it is sufficiently complex, and I have access to its source code (see Chapter 1 for an elaboration of this argument). To this, I must reflect on my bias as a researcher: I and my adviser Seth Cooper are developers of *Foldit*. While this stance is primarily for academic purposes, there is some inescapable amount of favor felt in wanting the game to succeed. I have tried throughout this research to remain impartial in my studies and elicit impartial, critical feedback from my participants; this statement is left here as a matter of transparency. Given how my findings agree with other researchers, and that these findings are not often to *Foldit*'s advantage, I would like to think that I have been mostly successful in distancing myself from this work.

In the next section, I describe how I continued the work of last chapter's SBCTA to design a new tutorial for *Foldit*. Afterwards, that tutorial was empirically tested in three phases. I describe my methods and results of each phase, and finally discuss overall results of implementing and empirically testing a new tutorial in *Foldit*.

8.2 Designing a New Onboarding

Exploratory, design-based research is complicated and messy. I would like to say that, based on the insights gathered from across this dissertation, I operationalized the lessons learned as feature changes directly. In actuality, not all of the work from the previous chapters was finished when I needed to design *Foldit*'s new onboarding experience. Even if they were, some insights are difficult to design specific features to address (such as in Chapter 5: CSGs have unclear gameplay and the developers are gatekeeping the development of community content creation).

Instead, the design process was more dynamic: as insights developed — as I received playtesting feedback, or heard from players outside of my experiments, or reviewed theory which provided a particular insight — I would build on my existing design concept to incorporate the insight or address the problem observed. And so, rather than paint a chronological picture of this process, the rest of this section describes the process narratively and thematically — I return to each insight gained over the course of this dissertation as organized into a handful of thematic concepts and describe the design process for addressing those concepts.

The design process was grounded in the ten steps of 4C/ID [540] and Guided Experiential Learning (GEL) [87, 85]. Using these theories, I prepared a list of learning tasks, a list of design principles to follow when creating new levels, and a template for level/lesson creation; see Appendix G for these materials.

As stated in the previous chapter, the design process was happening in tandem to the SBCTA study which produced 157 levels across 15 sets (henceforth 'worlds'). Each level had a learning objective, a description of the task, a list of prerequisite levels, and a recommended next level. However, after creating about 25 levels and partially implementing another 75, it became clear that this degree of detail was out of scope for the purpose of this dissertation. In the next revision, I focused on the first 3 worlds, which had a total of 27 levels. This subset of the onboarding included only three learning tasks: the first world targeted the core sub-task of late-game refinement. The second world focused on the prediction task: given a protein, predict its structure. The third world focused on the design task: given a chain of amino acids and some simple design goals, design a stable structure. These 27 levels were split into two modes, "Campaign" — the main mode which presents increasing challenges as it walks the player through learning the game, and "Tutorial" — a set of always-accessible guided scenarios which teach specific skills. The Campaign required Tutorial levels for progression and referred to them in the level selection screen (see Figure 8.1).

Although this subset of the onboarding does not teach all of *Foldit*'s mechanics, these 27 levels are already sufficiently complex to test any changes. For a summary of the levels designed, see Appendix H. In the section below, I will describe how the insights from this dissertation were empirically applied (or not), and why. A summary of the most extensive changes that were implemented or considered are highlighted in Table 8.1.

8.2.1 Player Background Matters

As the first insight gained, back in Chapter 3, is there anything we could do knowing that the player's background — their gaming experience, their prior knowledge, and their interests — matter a great deal in how they receive the tutorial? This insight relates to validated findings

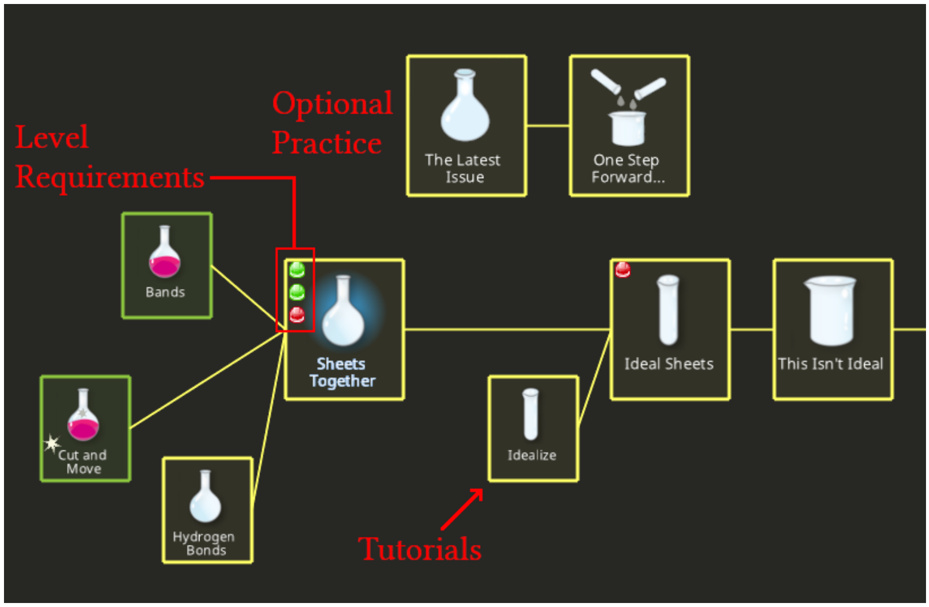


Figure 8.1: Campaign level selection screen. This image shows a subset of the second level set (or world). Tutorial levels are embedded into the level selection screen as smaller squares to denote prerequisites. Level requirements are also indicated as small status lights. Entirely optional levels are also visible on this screen. Completed levels are shown in green, while incomplete but unlocked levels are in yellow and locked levels are in red. Note that for testing purposes, all campaign levels were unlocked (shown in yellow), but in a practical deployment, levels of the Campaign would be restricted based on completion of prerequisites.

Feature	**Reasoning**	**Implementation**
FAQ panel	Smoother learning curve	Added
Campaign and Tutorial mode	Smoother learning curve	Added
Explanatory images	Smoother learning curve	Added
Minor polish changes	Polish for playability	Added
Main menu revision	Polish for playability	Added
Voice-overs	Dual-channel learning	Not implemented
Optional science information	Personalized learning	Cut
Background identification	Personalized learning	Cut
Move limits	Address “finnicky” levels	Cut

Table 8.1: Summary of the most extensive changes implemented or considered during development. These features draw from the insights described below. Some features were implemented and then cut based on playtesting feedback.

about personalizing gaming and learning experiences (e.g., [370, 371, 482, 43]), so there is strong evidence to suggest that *something* about this insight matters.

First, using standards and conventions is one way to lean into the player's background knowledge (see also Chapter 6 and [360, 310]); if a player has preconceived mental models for how to control a game, then we should use that existing mental model. In practice, however, the messiness of *Foldit*'s user input code combined with its non-standard gameplay made this heuristic difficult to fully adopt. I was able to make small adjustments, such as switching the usage of the middle-click and right-click to match user expectations. Furthermore, based on playtesting feedback, the scientific elements of the game cost the players much more cognitive effort than the controls, so additional changes to the control scheme were considered lower priority.

Second, I iterated on the use of optional scientific information — a feature originally tested in Chapter 3. Ultimately, many of these optional info-bites were cut during testing. Although they may have had a minor impact on motivation, in practice the scientific information created more confusion and breakdowns (cf. [227, 228]) and distracted participants from the primary learning goal (cf. the harm of secondary objectives [13]). From this, I believe that this scientific information is still valuable, but requires refinement: the info-bites being given (e.g., how a tool works) were too advanced for the novice learners and could have been saved for later, when the scientific information that would have been useful to them is more detail on the bigger picture of the scientific contribution model (see Section 8.2.5).

A third feature tested was allowing the player to explicitly identify their relevant background(s). I implemented a UI prompt which triggered once at the beginning of a player's first session and asked them to identify whether they are using *Foldit* for education, if they have a gaming background, and if they are knowledgeable in biochemistry. The prompt allowed users to check any number (including none) of these options and recorded their answers. Ultimately, though, this feature was also cut[1] because there were little to no practical applications of this knowledge. Places in the text which could have been adapted to a gamer's vocabulary or reference knowledge from a general biochemistry background were few and far between. In practice, more effort was put into refining the experience that everyone would see rather than trying to develop multiple experiences adapting to different player backgrounds.

Despite the difficulties implementing this in *Foldit*, I believe there is still utility in personalizing onboarding experiences, but with this distinction: novices should be given additional

[1] With one exception: the released version of *Foldit* now asks the player if they are playing for education, and if so unlocks all levels and adjusts a couple of other options which are suited for classroom use.

scaffolding while experts can skip past the additional tutorialization (cf. the expertise reversal effect [429, 222]). And for CSGs, they have two kinds of expertise: game mechanics and scientific mechanics. As shown in Chapter 6, successful games check the player's understanding using competence gates. Thus, in an ideal CSG tutorial, when it introduces control schemes, if a player is a gamer who understands or intuits the control scheme, they can quickly achieve the game's tasks and move on; players who struggle will instead receive additional scaffolding (and gamers will never see this additional scaffolding). Similarly, if a player understands the scientific meaning of a game, they can quickly move past the competence gates which check for understanding of scientific mechanics, while other players will receive additional tutorialization to teach about these scientific details. The reason this wasn't useful in practice for *Foldit* is because (a) gaming expertise does not help players understand *Foldit*'s gaming mechanics, and (b) few to no players tested had biochemistry knowledge which could have supported their understanding of the game, so this kind of expertise was not playtested or designed for.

8.2.2 The Giant's Staircase

The second major insight of this dissertation is what I called the "**giant's staircase**" [332] — the game's difficulty begins flat and trivial and suddenly spikes to insurmountable heights (see Figure 8.2). ECCSGs often have multiple giant's steps, such that even if you can overcome one hurdle, there are more remaining between intermediate and expert play, alternating between trivial and impossible.

The first step of this staircase is the entry skill barrier, identified in Chapter 5. Toward this, the new tutorial design included more explanatory images and signaling (e.g., pointing to specific virtual objects to guide attention, cf. [87, 10]). A second feature added to address the entry skill barrier was an "FAQ panel." This prototype feature shows one or more buttons that the player can click on to get additional information on-demand, but for content which is context-sensitive and relevant to their current level. For example, as shown in Figure 8.3, in the *Clashes and Voids* level, the player can click a button to get additional supporting information on voids, clashes, and which they should prioritize. In other levels, these buttons might offer tips, hints, relevant controls, or clarification on key terms mentioned in the main text (such as clashes and voids in this example).

Even after the entry skill barrier, the rest of the staircase is also a steep learning curve. From Chapter 6, we found that successful onboarding has a gradual increase in complexity, and furthermore that CSGs often suffer from "mechanic after mechanic" without sufficient practice (cf.

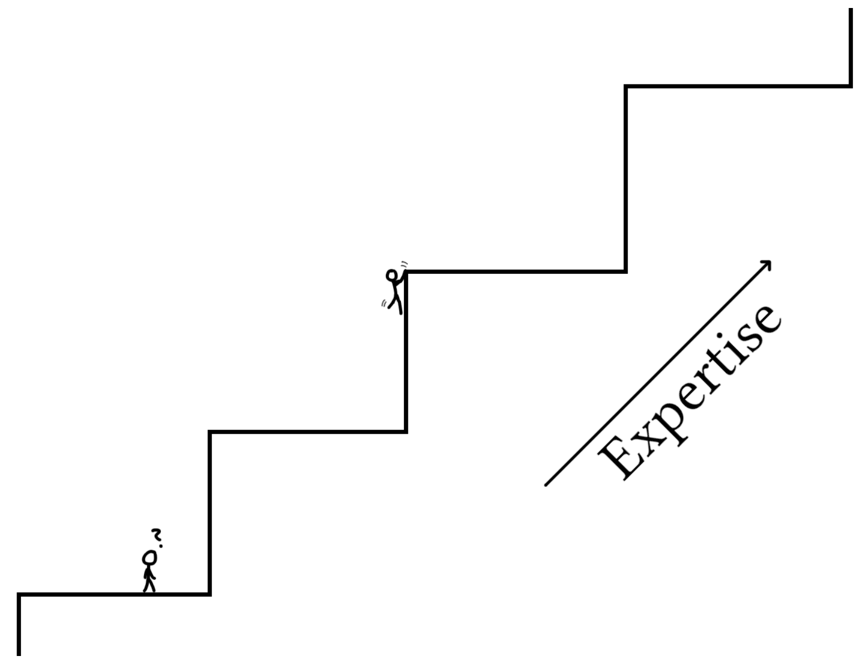


Figure 8.2: The giant's staircase. ECCSGs alternate between being trivially easy (a flat learning curve) and insurmountably difficult (a vertical learning curve).

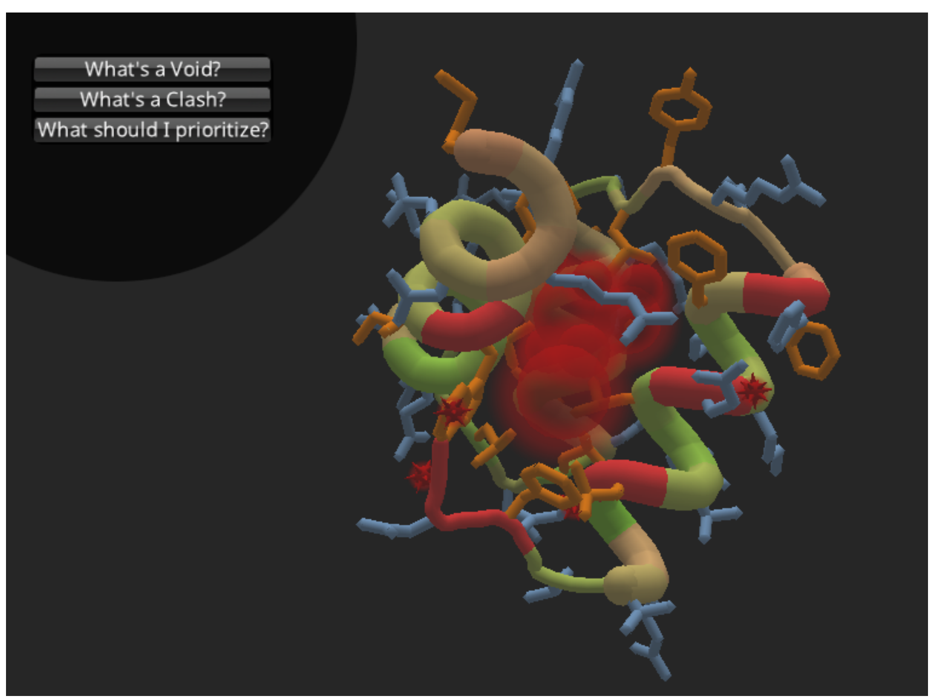


Figure 8.3: FAQ panel. This feature provides context-sensitive on-demand help. In this level, *Clashes and Voids*, the player can click a button to get additional supporting information on voids, clashes, and which they should prioritize.

the principles of applying knowledge and varying practice [272]). Taken together, these insights suggest that the onboarding should use lots of practice to gradually increase difficulty and complexity. To satisfy this heuristic, the new design tried to add additional practice levels (recall that the first iteration of the design had 157 levels, compared to the original experience which has 34 as of early 2023). Moreover, the new design employs what could be called a "trunk and branches" level structure, such that there exists a main path (the trunk) which introduces new mechanics with a small amount of practice and a lower level of difficulty, whilst optional paths (the branches) provide additional practice and greater difficulties. Moreover, the branches may include real science puzzles or secondary mechanics that are not necessary to learn in order to play, but empower expert players who are willing to explore the depths of *Foldit*. These branches were meant to address the insight from Chapter 7: encourage practice and learning beyond the tutorial. Although a couple of optional levels were implemented, most playtesters did not engage with them, given that (a) they were never recommended by the game system, and (b) they were intentionally more difficult levels.

In retrospect of the level structure, I still was not able to add as much practice as I would have liked. Playtesters still complained about the constant introduction of mechanics and even suggested that they would have preferred a longer tutorial with shorter puzzles. To this, I considered implementing a feature which would allow a level to be a sequence of mini-levels, but due to technical constraints and scope limitations, I was not able to achieve this. The fact that additional practice was infeasible to implement speaks to a finding from Chapter 6: the suggestion of simplifying mechanics. If *Foldit* had fewer mechanics to teach, there would be more room in the scope of design to add additional practice and smooth out the learning curve. However, simplifying *Foldit*'s fundamental mechanics (a massive undertaking of code and design) while scientists were still trying to use those mechanics for their own research was absolutely impractical.

Another insight which relates to the giant's staircase is to pace learning and check understanding (Chapter 6). To this, the Campaign levels along the "trunk" were designed with the intention of being competence gates, i.e., being completable only through the demonstration of skill mastery. In practice, some players were still able to complete the levels without understanding, and other players were completely stuck because of their lack of understanding. The former issue is partially due to *Foldit*'s "finicky" level design (see Chapter 5). Addressing the latter issue was a matter of providing additional scaffolding, such as the FAQ panel from Figure 8.3 and additional explanatory images, such as in Figure 8.4. I wanted to additionally add explanatory animations (following the principle of dynamic visualizations [539]), but this proved to be technically difficult in *Foldit*'s code.

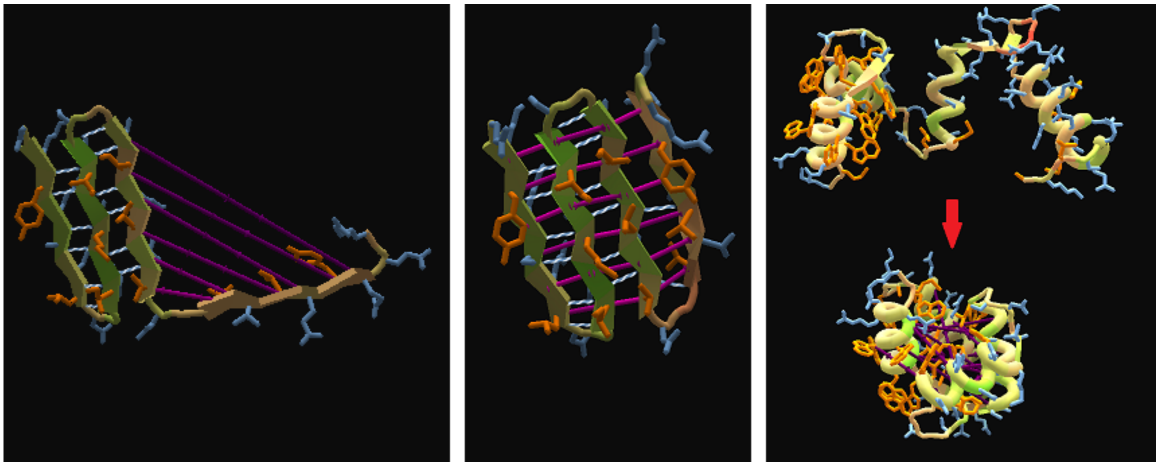


Figure 8.4: Examples of explanatory images. These images are presented to the player when appropriate to help teach them about rubber band usage (the purple tubes which apply a contracting force). Left: a hand-holding guide for how to use bands to solve the specific level that they are on. Center: a generic example of using bands to hold sheets together. Right: a generic example of using bands to pack a protein closer together.

Additional scaffolding was also aimed at addressing two findings from Chapter 5: unclear gameplay and a lack of feedback. Feedback has long been recognized as a critical part of both games and learning [539, 445, 247, 151]. In *Foldit*, although the game provides quantitative feedback on your immediate action (i.e., whether your action provided points in the short-term), it provides no qualitative or long-term feedback, which is important for a game about creative problem-solving and slow, "system 2" thinking [248]. Yet, this was a feature which I was unable to implement due to scope: how could the game provide dynamic feedback on whether the action a player just took was ultimately helpful toward the long-term objectives? Even when providing a "guide" of what the protein should look like, players struggle to compare their current action to whether they are achieving long-term success. The challenge of qualitative feedback recalls two insights from Chapter 4: the importance of choosing a CSG's abstraction and gamification. That is, how one frames the task can make a big impact on how intuitive the task feels and how feedback is interpreted. For example, most games today have a concept of fighting monsters and losing health; this design metaphor now makes for easy interpretation — if you lose health, that's bad; if your enemy loses health, that's good. In *Foldit*, on the other hand, you can gain points by putting your protein in a worse position long-term. We can imagine an alternative where instead of having points at all, *Foldit* puzzles simply have a set of objectives such as "15 orange segments in the protein's core" and "0 clashes" and "at least 1 helix and 3 sheets." This framing — positioning the qualitative, long-term objectives as the goal while eschewing the gamification of smaller details — could help players focus on the larger, core gameplay loop. This would in turn smooth out the aspect of the learning curve where players focus too narrowly on making points go up and get stuck on a puzzle where short-term gains are insufficient.

Next, in Chapter 4 we identified that players are concerned with task quality. For *Foldit*, this means matching players to levels of an appropriate difficulty. One possible solution here is a dynamic level recommender which would match players with levels based on their past demonstration of skills. Although no such feature was implemented for this study, my past work with Stoneman and Cooper on *Foldit*'s Dojo mode [491] is an early prototype of this concept. Future work could further explore modeling players' skills to match them with tasks of appropriate difficulty (see also Sarkar's work, [449, 447, 448]).

Lastly, the latter steps of the giant's staircase are steep because the tutorial ends instruction early (Chapter 5). Without further instruction, moving from tutorial levels to scientific play is extraordinarily difficult for players [332]. Therefore, one of the goals of the new design was to cover all mechanics. Indeed, the original design of 157 levels in fact covers all of the mechanics in *Foldit*. However, as noted, for the purpose of empirical testing, I refined the first 27 levels of this design. Thus, while the insight of teaching all mechanics was out of scope for the purpose of empirical testing, I still recommend this to other CSG developers.

8.2.3 Communicate, Communicate, Communicate

A major trend across most of the studies in this dissertation was an emphasis on clear, frequent communication. In Chapter 4, scientific communication was identified as a critical problem both by players and other stakeholders. In Chapter 5, we noted (agreeing with prior literature) that for a player to feel motivated, they need both to understand their scientific contribution and to gain something from play for themselves.[2] In Chapter 6, we identified that CSGs don't explain their scientific goals sufficiently.

Rather than being an instructional design issue, this insight relates to a problem in citizen science [439]. Whilst there exist heuristics for implementing clear, frequent communication, the issue is one of practicality, given that scientists and the rest of the CSG development teams are neither trained nor funded for this kind of public outreach (see Chapter 4).

[2] Here, that self-gain is mostly a novel, educational experience. As noted in Chapter 1, this dissertation focuses primarily on the teaching goal of onboarding rather than the motivational goal, which is why I did not put further effort into implementing further self-gain features.

The problem of communication is multifaceted. The above issues describe the communication between scientists and players, but there are also communication breakdowns between scientists and game designers (see Section 4.2.2.4). Moreover, as shown in Figure 4.3, there are multiple vectors of communication involved in publicizing scientific knowledge. Even the communication between players is important for their social learning (cf. Chapter 5).

Yet, a re-design of the onboarding experience can't address these deeper, systemic issues. I did try to implement optional scientific knowledge (see Section 8.2.1), but this was ultimately more distracting than helpful as implemented. During this time, other paratextual scientific communication was implemented too: I published weekly *Foldit* newsletters for two years and one of the lead *Foldit* scientists released a monthly video blog. However, both I and the scientist are now leaving *Foldit*, and the future of these communications are unknown. This echoes a sentiment from Chapter 4: developers take charge on particular sub-projects, but when they leave, the sub-project is abandoned.

In an ideal scenario, the scientific onboarding — that is, onboarding players to understanding the scientific contribution model and how the CSG gameplay contributes to scientific knowledge production — is achieved before the player needs to put cognitive effort into learning how to play the game. If the player learns how the game operates on a basic level from (1) the advertisements and marketing material, (2) the game's website or store page (in the case of being hosted on game distribution platforms), and/or (3) the game's narrative introduction (even if that narrative is non-fiction!), then the player could understand the "big picture" of the game (cf. Section 8.2.5) without splitting cognitive attention between learning game mechanics and understanding the purpose of the game.

8.2.4 Polish For Playability

Another cross-cutting theme was the simple fact that CSGs are not good software. That is, in multiple studies (Chapters 4, 5, and 6), developers, researchers, and players all came to the conclusion that CSGs have technical bugs, software issues, unintuitive interfaces and controls, and user experience (UX) issues — and these issues severely impact the player experience. To clarify, as supported by prior literature [12], what harms the player experience is not that the game is unpolished in its art or music, but that issues with the user experience distract and disrupt the player, taking away cognitive resources by requiring them to try to make sense of the issue and resolve it for themselves.

Toward addressing this, the development of the new onboarding experience included a variety of minor tweaks to assets, wording changes, code fixes, and so forth — such as the ability to add labels to elements which were not originally coded to do so so that a tutorial could gesture to that particular thing. Although these changes are difficult to explain succinctly, playtesting suggested that they contributed to the overall UX and quality-of-life for players. Yet, there were still issues which remained unaddressed due to complexity with the underlying code. For example, when a player adjusts the Clashing Importance (a slider between 0.00 and 1.00), they can choose to manually type in a value. However, if the player does not press Enter after typing in their value, the window will *show* the value they typed in but *use* the original value. During playtesting, this created a lot of confusion for players; yet, when I investigated addressing this fix, complications in *Foldit*'s code made corrections infeasible for the scope of the study. Players similarly suggested many quality-of-life improvements to me during playtesting, and while I was able to address some of these, many more were impractical when considering the costs and benefits.

The issue of having many bugs to fix recalls the suggestion from Chapter 6 of simplifying mechanics: if *Foldit* were overall reduced in scope, there would be fewer features which could break or need improvements. Furthermore, as described in Chapter 4, ECCSGs become bloated with features in this way because the development teams are constantly adding new features rather than polishing the existing ones, due to how scientific funding operates. There is no simple answer to the challenges described here, but starting from a simpler scope — and carefully designing one's abstraction so that the game can, without large changes, tackle a range of scientific problems — could ameliorate this problem to some degree.

8.2.5 The Big Picture

Multiple insights fall under the category of teaching the player about the "big picture," or the core gameplay loop and the scientific contribution model. In Chapter 5, this was about teaching the whole task rather than taking a part-task approach. In Chapter 6, this was about setting expectations and teaching players how CSGs contribute to science. And in Chapter 7, this was literally "give the big picture up front." These insights largely mirror the approach of 4C/ID, which argues for a whole-task-oriented instructional design [537, 536, 538, 542].

Therefore, the new onboarding experience aimed to focus on the whole task — the big picture — and set expectations appropriately. As described earlier, levels were divided into worlds. Part of the intention of this design was to provide an overview of the game structure: players could see the progression of worlds as a zoomed-out look at their tutorial progress. Moreover, the first world was designed to teach the core gameplay and scientific contribution model early on. The first world was the level set most iterated on, and it eventually focused on introducing the core play of *Foldit*, basic controls, and the two most fundamental tools and rules of *Foldit*.

In this way, the organization of worlds also aimed to "Reinforce the intended structure of knowledge" (Chapter 7). The visual design of the tutorial levels leading into particular challenges tried to show the players how each tool or concept applies to the upcoming challenges, and how all of the practice levels funnel into the larger scientific tasks (Prediction and Design puzzles).

Moreover, drawing again from Chapter 7, I aimed to "Situate learning in applicable, meaningful contexts." Campaign levels were designed with a focus on whole tasks: rather than practicing just a new tool, the player was practicing a new tool in the context of the gameplay loop they have already learned. Additionally, some campaign levels mimicked real science puzzles to further reinforce orientation toward scientific tasks. Given the insight "Don't separate the tutorial and the game" (Chapter 6), in the ideal case I would have used actual scientific puzzles and dynamically inserted active puzzles into the Campaign as practice; however, this was technically infeasible for implementation.

8.2.6 We Play in a Community

Some of the insights discovered in this dissertation don't relate directly to the single-player onboarding experience, but rather to how players are inherently part of a community of learning and practice (cf. [180, 555]). In Chapter 5, I noted that players trying to gain expertise feel low knowledge self-efficacy, a low sense of belonging, tensions from the competitive aspects of play, and that the creation of community content was suppressed by developer gatekeeping. Unfortunately, I was unable to find ways to address these insights from the onboarding experience. These issues are systemic to the game itself and the relationships between the development team and its players, and while I did take personal action to addressing these issues for *Foldit*, that action was not an empirical part of this study.

In Chapter 7, one recommendation was that developers should "Embrace social learning and paratext use." Although also outside of this empirical testing, there are two features of note which I implemented for *Foldit*. First, when a player is stuck on a tutorial level for a set length of time, they are offered a link to the fan wiki page for that puzzle, which includes various walkthrough support.[3] Secondly, I created a Discord server[4] for players to congregate and discuss; the server is also linked to the in-game chat through an IRC relay bot.[5] As of early 2023, the server has 2,775 members and hosts regular chats with the developers through scheduled "Office Hours" events. However, because these features are outside of the onboarding experience, they were out of scope of empirical testing. Although I was unable to develop onboarding features which address the social barriers, community is a critical aspect of any citizen science project [383, 376, 269, 240] and future work should continue investigating how best to promote social learning. This brings us to the last group of insights, social learning as part of a larger cycle.

8.2.7 Reveal, Review, Repeat

In Chapter 5, we found that the process of gaining expertise is a cycle of exploratory learning followed by social learning. Zooming into the exploratory half, in Chapter 7, we suggested that developers design for discovery and self-reflection. These findings relate to Gee's "cycles of expertise" [178], Jennett et al.'s MLC model of engagement and participation [241], and Squire's case study of Apolyton University, an online gaming community with a learning focus [486]. More generally, then, this cycle of learning is about discovering information, integrating it into one's mental model, and iterating (cf. elaborating on a mental model, as Reigeluth's Elaboration Theory [426], and integration of knowledge as part of Merrill's First Principles of Instruction [325]). Hence, "Reveal, Review, Repeat."

As noted in the last section, social features are difficult to implement in an experiment setting, but when players were stuck, they were shown a link to a fan wiki page with walk-throughs. Similarly, in the new onboarding experience, when a player was stuck, a visual guide would appear which shows them an outline of one way to solve the level. The FAQ panel further helped with this approach: players could (and did, as playtesting showed) experiment with the level before asking the game for hints.

If I had more time to iterate on the design, I would have refined the FAQ panel to be even more assistive in this regard, as well as adding tooltips on mouse hover (cf. the praise of tooltips in Chapter 6) to support players' exploration. Perhaps more could also have been done to prompt players to join a group and discuss with their team members about their progress; however, testing

[3] This could not be done for the new onboarding experience, since the puzzles were freshly created and had no fan wiki pages.

[4] `https://discord.gg/Ffgx2KJ`

[5] Thanks to `https://www.npmjs.com/package/discord-irc`.

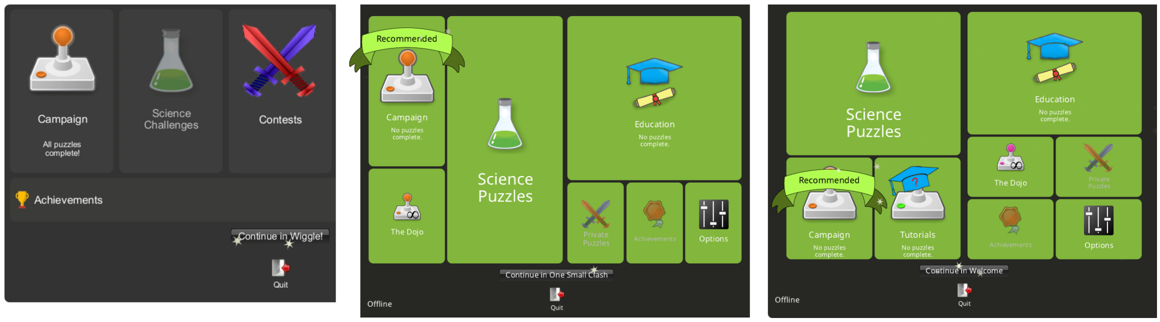


Figure 8.5: *Foldit*'s main (mode selection) menu. Left: an excerpt of the original mode selection menu which displays each mode equally. Center: the revised menu which displays the modes in a variety of sizes to show importance and marks the Campaign as "Recommended." Right: the experimental version, which separates the Campaign and Tutorial.

such a feature would involve a longitudinal social study which was out of scope for this chapter.

This section concludes the seven themes of insights gathered across the last few chapters. Now that I have described the features I was able to implement — with some discussion of why I was or was not able to address different insights, and what prior literature or theories each feature draws from — I move onto a few final thoughts on features which were considered based on various theories of cognition. Afterwards, I wrap up with a few general reflections on this design process and outcomes.

8.2.8 Theory-Driven Features for Guiding Cognition

Although not driven by findings from my own studies, several features were considered based on the learning theories I am familiar with and playtesting results which relate to those theories. For example, tutorial text in *Foldit* jumps around the screen as it points to different on-screen elements (cf. signaling [539]). Players spend significant perceptual effort to locate this text, especially when players are using working memory to learn (cf. [116, 118, 285]). Thus, one feature considered was to fix the on-screen text to a specific position on the screen, though this proved to be technically infeasible. Similarly, rich text was considered to guide player's perception through the use of bold lettering or colors, but this was also technically difficult. However, I was able to implement a revision of the main menu screen to guide perception. As shown in Figure 8.5, the new mode selection screen uses variations in size and a "Recommended" banner to draw player's attention to the more important game modes.

Another feature considered was adding voice-over such that verbal information was presented audibly and non-verbal information was presented visually. This would enable dual-channel processing, allowing the player to learn better via the modality effect [318]. However, this feature was cut for scope — both because the verbal information was constantly being revised and because the financial cost of hiring a voice actor exceeded my research budget.

Although the details are out of scope for this chapter, the final onboarding design was also guided by the principles from 4C/ID [539] and GEL [87] in Appendix G.2. This included, for example, prior knowledge activation (e.g., recalling a tool the player learned about), segmentation (splitting instruction into meaningful chunks via the separation of levels and sets), worked examples (e.g., showing examples of well-folded proteins), and backwards chaining (teaching a complex procedure from the last step to the first). Future work could consider also examining the instructional design principles from [272] and investigating how easily these principles can be operationalized and implemented in onboarding design.

8.2.9 Design Reflections

In summary of the design process, I created the tutorial first top-down, then bottom-up. That is, I designed the curriculum, the progression, and the levels themselves iteratively using my own design skills combined with the instructional principles in Appendix G.2, and then I adjusted the levels iteratively based on playtesting feedback. Whilst I tried to address all of the insights gathered in this dissertation, there were two major barriers to doing so.

First, the technical barrier: *Foldit* is, at the time of writing, nearly 15 years old, and it has had dozens of programmers (many of whom were students) working on it over the years. Moreover, most work on *Foldit* has been feature-driven — as described in Chapter 4, CSGs are funded by grants focused on specific scientific outcomes. These factors combined meant that *Foldit*'s code was particularly inflexible to changes, especially anything which tried to change the overall user experience. For this reason, some insights which seemed simple — such as boldfacing key terms in the text to guide perception — would have taken weeks to months to properly implement. This reflection speaks to how important it is to think through these design issues from the beginning while the code and the game design are still malleable.

Second, relatedly, some insights couldn't be implemented because the issues involved were systemic to the game's mechanics or affected the player community in ways that an experimental version of the tutorial could not address. These problems likewise could have been addressed at the beginning of *Foldit*'s design but became harder to uproot as the game and community developed together. For example, new players of *Foldit* empirically complain about not being able to learn from other players — a side-effect of competition (see Section 5.3.4.3) — but taking the competitive aspect out of *Foldit* now would require years of developer-hours.

Suppose that a CSG developer wanted to copy my approach from earlier in their design process: should they? What pitfalls should they avoid? I believe that my approach (i.e., designing top-down, then bottom-up, according to the insights and instructional principles cited) was overall a helpful one for guiding design; yet, it was easily over-engineered. In the first iteration using SBCTA, I produced 157 levels. Although these might have been helpful for comprehensively teaching players, I think the design lost track of enabling players to perform the whole-task from early on. Similarly, it was easy to get "tunnel vision" or become narrowly focused on addressing problems that came up during playtesting. When observing a player, their issues were more noticeable and thus felt more urgent (cf. the availability bias [526]).

As a detailed example of how this tunnel vision hindered development, one feature I spent a lot of time developing (and ultimately cut) was move limits (cf. [173]). I noticed a problem during playtesting: players would get themselves into a losing game state and not understand how to revert their actions to solve the puzzle. Unlike *Sokoban*-like puzzle games such as *Baba Is You* [212], the failure states in *Foldit* are softer — it is not impossible to fix one's mistakes, just harder. This becomes a design challenge — how do you help the player recognize and recover from failure states? Move limits were an attempt to address this challenge by forcing the player to reset if they've spent too many moves: because each level could be solved in a few moves, if a player spends more moves than is necessary, they are likely in or approaching a failure state. In practice, however, move limits suppressed exploration and only made the game even more 'finicky' than it already was. Move limits addressed a symptom rather than a cause: players were getting into failure states because the levels were designed as puzzles (with a single solution) rather than problems (which can be solved in various ways). Therefore, this feature was cut. Instead, I focused on addressing the underlying cause and re-designed levels to be less of a puzzle and more of a problem. There are still soft failure states, but on-demand and dynamic help is now offered to suggest that players undo or restart if they find themselves stuck.

Now that I have described the design process, how effective was the new onboarding experience? The rest of this chapter details the empirical testing conducted, which turned out to be (as many exploratory, qualitative studies are) very messy. In the next section (what could be considered 'Phase 0'), I describe the process of iterating on the design through live playtesting to polish out any major bugs or issues (e.g., this phase is where I noticed that move limits were not working). Once the design felt finalized, I moved onto Phase 1 of empirical testing, in which I randomly assigned participants to play, at their leisure, a version of *Foldit* so I could compare the impacts of instructional design methods. However, the results of this study were too noisy to make substantial claims, so in Phase 2 I returned to an iterative, exploratory, "live" approach toward discovering why Phase 1 participants were experiencing breakdowns when playtesters were able to complete the onboarding. Finally, in Phase 3, I combine these approaches toward one last attempt to measure the impact of the new onboarding material.

8.3 Iterative Playtesting

Prior to empirical testing, I conducted a design-based exploratory study for iterating on the development of the new tutorial. The goal of this phase of research was to ensure that the new tutorial design had a smooth player experience and to address any bugs or issues identified by players.

8.3.1 Methods

Five participants were recruited online through emails to mailing lists and posts on social media. Due to convenience, recruiting especially targeted local populations such as the local game development industry and Northeastern University students, staff, and faculty.

Participants attended a one-hour online session (scheduled at their convenience) where they played a version of *Foldit* with minimal guidance from the researcher (for example, I asked participants to move on if a level had an issue making it impossible to complete). After the session, participants were invited to return for another session if there was further testing required. In this way, I conducted eleven sessions across the five participants. Because development was iterative, not all participants received the same version, and participants who attended multiple sessions typically saw multiple versions of development. In total, there were seven iterations of the design.

For all phases of this study, participants were required to be 18 years old, have not played *Foldit* before,[6] and own a Windows computer, since the experimental version was developed only for Windows. Methods (in all phases) were approved by Northeastern University's Institutional Review Board. All participants (in all phases) were remunerated for their participation with a $15

[6] One participant in Phase 3 admitted to playing before but they played little and long ago enough that they did not remember anything and were effectively new to the game.

USD Amazon gift card for each session of participation (some participants in early phases attended multiple sessions).

8.3.2 Results

Most of the results of this study were iterative changes, such as copy editing, changes for clarity, and adding more explanatory images. For example, as described in Section 8.2.9, the move limits feature was implemented and then removed. It was also during this phase which I implemented the FAQ panel (Figure 8.3) to embed context-sensitive help. Some of these changes were added to help orient the players toward their goal since the goal state was not always clear. During testing, participants often got caught up on minor UX issues: the camera zoom was inverted to their expectations, or the camera moved too slowly, or the levels were finicky, or it was difficult to navigate spatially, or they didn't like the interface, and so on. Similarly, participants were sometimes confused when the text mentioned "this" or "here" without better signaling — where possible, more clarity was added to guide players' attention and perception in these cases.

Participants were also burdened by the tutorial being overall cognitively onerous and lacking global feedback — participants were unsure if they were "doing it right." They mentioned a lack of understanding the "why" behind mechanics; yet, when presented with more scientific background on why certain mechanics were what they were, this information largely did not seem to help them play the game better; if anything, it distracted them from focusing on the "how" of using mechanics. Similarly, when levels were guiding them via hand-holding, participants would sometimes say "OK" to everything and follow along but not truly understand; although some competence gates were added to catch and prevent this issue, sometimes players would pass a section of gameplay without integrating the concepts taught into their mental model, causing breakdowns later in play when those skills were prerequisites to more advanced learning.

Moreover, participants were frustrated by the constant introduction of mechanics. Despite adding more practice levels than the original tutorial, the amount of practice was still insufficient for mastery of the mechanics. One participant even asked for a longer tutorial but with shorter puzzles, i.e., for a more gradually-paced onboarding. This was something I tried to design for, but for the purpose of empirical testing, there was a balance to strike between adding levels and recruiting participants to play all of the levels. Additionally, as the level designer, it was difficult to construct problems that were neither trivial nor impossible and also required the mechanic that the level was intended to introduce or practice. This problem highlights one of the systemic issues that I could not address — *Foldit*'s mechanical design was built with many scientific concepts in a way that muddies intuitive gameplay. For example, *Foldit* prominently features a score for your overall progress on the level; yet, all veteran *Foldit* players understand to ignore this value until nearly the end of the level because one's score will go down before it goes up (cf. Chapter 7) For novice players, this is unintuitive, and they chase the feedback given by the game which tells them that they are doing good by getting a few points here and there. In designing levels that would successfully teach new mechanics, I had to implement new features such as additional objectives to help guide the player or influence the score to encourage the desired output.

Another notable observation is that the participants' state of mind mattered a lot (cf. their background matters, Section 8.2.1). When participants were intimidated, confused, or even simply tired, they would skim the text rather than reading for comprehension, and thus understand less and less in a positive-feedback spiral. On the other hand, when participants approached the game inquisitively, they would continually reset the level and try again, commenting how they understood a little better with each attempt.

8.3.3 Discussion

Similar to Chapter 6, this playtesting demonstrated that minor breakdowns (caused by technical or UX issues) create friction which hinders the learning process. The players' frustrations weren't surprising, but they were difficult to address. Opportunities within *Foldit*'s inflexible code for providing meaningful, immediate, qualitative feedback were difficult to identify. In the same way, I struggled to create effective competence gates — levels that were neither finicky nor trivial and completable if and only if the player demonstrates an understanding of a concept. Thus, if there are any takeaways from this study, it is that design was made much more difficult by *Foldit* having inflexible code and mechanics. By the end of testing, however, players were able to complete all levels and only minor or unaddressable issues were being raised, so I moved onto empirical testing.

8.4 Phase 1 Empirical Testing

In the first phase of empirical testing, research was driven by the question "How does the new onboarding experience compare to other forms of instructional design?"

8.4.1 Methods

Participants (n=32) were recruited online through emails to mailing lists and posts on social media. Due to convenience, recruiting especially targeted local populations such as the local game development industry and Northeastern University students, staff, and faculty, as well as contacts known to the researcher. During analysis, four participants were removed for a failure to comply with instructions, such as by responding "[...]SFDSFSDFSDFSDF[...]" when asked to write about their experiences in detail. Minor compliance mistakes were allowed, such as not uploading their work in step 4 (see below). Thus, 28 participants remained in the final dataset (16 male; 10 female; 2 non-binary; $M_{age} = 29.8$; $SD_{age} = 13.0$).

8.4.1.1 Protocol

All participants were asked to follow this set of instructions from their personal computer at their own convenience:

1. Create a *Foldit* account.
2. Download *Foldit* (either the default or experimental version).
3. Complete a task which varies by experimental condition.
4. Play a custom science puzzle until you have reached the highest score you can achieve; upload the solution to the researcher.
5. Complete a post-experience survey.

The custom science puzzle replicated an early version of *Foldit*'s Coronavirus puzzle series. In the custom puzzle, players are given an extended chain of isoleucines (effectively, a blank slate of a protein) and asked to create a protein which binds to a segment of the SARS-CoV-2 spike protein.

In the post-experience survey, players were asked to open-endedly describe their overall thoughts and feelings on *Foldit*, the best and worst aspects, and whether they would continue playing (why or why not). Then they were asked about the skills they learned and the difficulty of the game, copying the questions exactly from the first study of Chapter 4 (see Appendices B.1.3 and B.1.4). Next, participants were asked three quiz questions to test their knowledge and understanding of the game (see Appendix I). These questions were checked with a *Foldit* expert player (named Susume) to ensure that the expected answers are accurate and knowable. Then participants were asked to estimate how long they spent on the learning task (step 3) and how long they spent on the scientific puzzle (step 4). Finally participants were asked to fill out demographic information on their age and gender and given space to describe any issues they encountered during their participation.

8.4.1.2 Experimental Conditions

Participants were randomly assigned to one of four conditions. In the `Control (C)` (n=10), the experimental task was to play the original tutorial for a duration of their choosing between 10 minutes and 6 hours. Likewise, the `New Tutorial (NT)` (n=5) condition was asked to play the experimental tutorial (the version of the game iteratively designed as explained in previous sections) for a similar duration (10 minutes to 6 hours). To try the `NT` version for yourself, the Windows executable is provided at `https://osf.io/rmext/`.

The purpose of the third condition, `Cognitive Apprenticeship (CA)` (n=7), was to compare game-based learning to a non-interactive medium. In this condition, participants were asked to watch at least 10 minutes of a 53-minute video[7] in which I perform a walk-through / talk-through of the New Tutorial. That is, I recorded myself playing through all of the New Tutorial, reading the text out loud and talking through my actions. This condition is called cognitive apprenticeship because it was intended to teach via cognitive apprenticeship; that is, by making my (sufficiently expert) thinking explicit and demonstrating the correct mental model for approaching gameplay [92, 93]. In this way, participants in this condition should "complete" levels without the frustration of failure, but lack the learning aspects of hands-on practice.

The purpose of the fourth condition, `Video Instruction (VI)` (n=6), contrasts the previous three by having no reference to tutorial levels at all. Participants in this condition were asked to watch at least 10 minutes of a 19-minute video[8] in which I explain the concepts of *Foldit* explicitly but without any tutorial levels providing goals or context. Instead, the examples shown on screen are in a "sandbox" level or taken out-of-context from various puzzles. Therefore, this condition is also a non-interactive medium, but further removes the aspect of level-based learning and uses a more traditional method of instruction (i.e., lecturing). Although the efficacy of the Video Instruction was not verified rigorously (beyond informal checks with another *Foldit* developer), the script roughly follows the instructional material of the tutorials.

In this way, conditions `C` and `VI` provide two kinds of baselines: traditional gameplay and a traditional learning environment, respectively. The `NT` condition is the primary experimental group (testing the design principles and insights explained in the previous sections), whilst the `CA`

[7] `https://www.youtube.com/watch?v=BjrUJTAngIk`
[8] `https://www.youtube.com/watch?v=XP1iaGrzfCI`

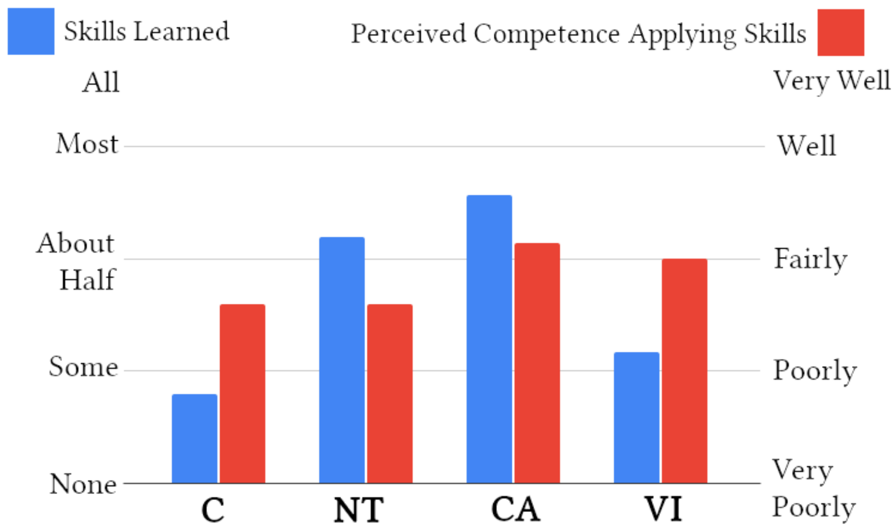


Figure 8.6: Participant reports about the skills learned in *Foldit*. Each question was presented as a 5-point Likert scale (coded as 0–4). Standard deviations for each bar from left to right: 0.42, 1.17, 2.17, 1.52, 0.98, 1.09, 2.23, 1.35.

condition measures the difference between *watching* the experimental onboarding be completed and *experiencing* it for oneself. Previous empirical evidence suggests that watching gameplay can provide a "vicarious experience" and trigger the worked-example effect [380]. Given the exploratory nature of this work, only descriptive statistics are reported in the results.

8.4.2 Results

Participants overall reported understanding slightly less than half of the skills needed to play, as shown in Figure 8.6. This is approximately the same result found in Chapter 4, so we cannot conclude that any condition made a significant impact here.

Next, participants reported (after the experience) on how difficult each section of the experience was: the beginning, middle, and end of their learning, and the science puzzle (see Figure 8.7). Notably, the `CA` condition started the most difficult, while the `C` and `NT` conditions ended the most difficult. However, given the large variance in responses, this again is not conclusive of any outcomes. With respect to the quiz of their understanding (see Appendix I), very few participants successfully answered any questions. No participant answered question 1 correctly. Question 2 was correctly answered by two participants in `VI` and one in all other conditions. Question 3 was correctly answered by only one `VI` participant (who also answered Q2 correctly).

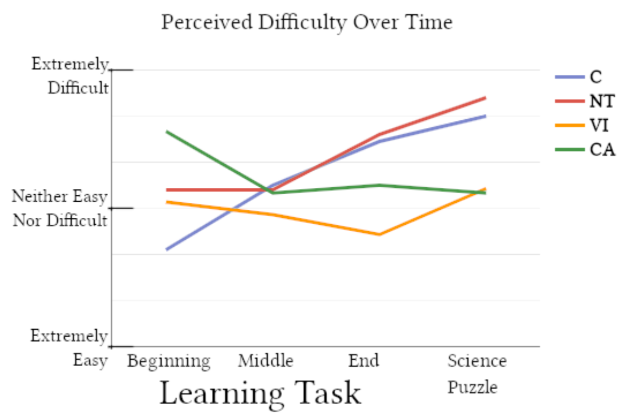


Figure 8.7: Participant reports of perceived difficulty over time, as a 7-point Likert scale from Extremely Easy to Extremely Difficult (coded as 0–6). The CA condition started the most difficult, while the C and NT conditions ended the most difficult. Standard deviations from left to right: (C) 2.28, 1.97, 1.46, 0.82; (NT) 3.13, 3.13, 1.95, 0.90; (VI) 2.34, 2.25, 1.22, 2.07; (CA) 2.67, 2.27, 2.37, 1.99.

Next I evaluated participant reports of time spent playing. This analysis could only be approximated because participants varied in the detail they gave (e.g., “20-30 minutes,” “3 hours”). For the learning task, NT participants seemed to spend the longest (approximately 60 minutes) while other conditions spent approximately 20 minutes. For the science puzzle, participants seemed to spend around 30–60 minutes, with C participants on the low end, CA participants on the high end, and the other participants roughly in the middle.

For the open-ended questions, I coded participant responses using the same codebook as [333] (Chapter 4). Yet, no clear themes or even trends could be identified from their responses, with a few notable exceptions. First, participants in all conditions were confused and frustrated by the game. They reported that their favorite moments were the ‘aha’ moments when — if ever — they briefly understood a mechanic or their score increased. This is exemplified by one participant who wrote “I thought for a fleeting moment that I understood what I was doing as I increased my score, and then realized that I had no idea. That brief moment was nice, though.”

Participants in the VI condition felt daunted by the video. Some said that they would have had an easier time if they started with the tutorials first. Others noted that the video scared them away or lured them into a false sense of competence which was shattered when they started the science puzzle.

In general, participants wanted more time to practice and understand the material. They

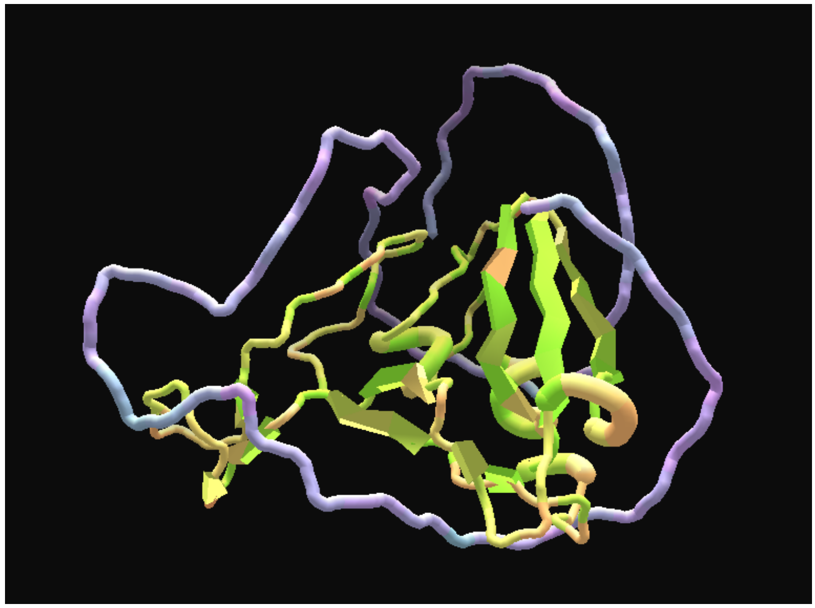


Figure 8.8: The highest scoring solution from a Phase 1 participant. Their fold (purple) wraps around the spike protein (green) without any secondary structure. This solution is a premature optimization of points and would not fold properly if synthesized in a lab.

wished for fewer software issues and the experience to generally be clearer and less frustrating. As found in Chapter 4, participants also commented on appreciating the educational aspects and the gamification, but felt that the experience had a steep learning curve which was confusing and difficult to control and interact with.

In assessing their performance on the science puzzle, I tried to compare participant scores and solutions. However, little information could be determined from their scores because no participant created a meaningful solution. Shown in Figure 8.8 is the highest-scoring solution from any participant — this fold merely wraps the original extended chain around the spike protein and does not attempt to form any secondary structures. Other participants similarly submitted only the extended chain pulled slightly and wiggled. This output is likely due to the fact that participants did not get far enough into the game to learn the techniques of designing proteins which are required for solving this puzzle.

Lastly, to summarize issues that participants reported, some players noted installation issues, freezing, crashing, lagging, slow and frustrating controls, and difficulties navigating to the science puzzle. These issues broadly comment on the usability and user experience of *Foldit* (cf. Chapter 6) but does not indicate any other problems with the study design or differences between conditions.

8.4.3 Discussion

In the first phase of empirical testing, we were unable to draw any conclusions regarding the efficacy of each condition. Based on the duration participants reported playing, I expect that this is because they did not get far enough into the game (or videos) to see differences emerge between the conditions. Moreover, because participation was allowed to be at one's convenience, I could not control for how much attention participants actually paid to the learning material. They could have been distracted or participating after a long day of work, for example.

Notably, it was very difficult to recruit participants for this phase because of how many suspected bots and malicious accounts were attempting to sign up for this study, since it was posted publicly in a way that invited the attention of automated survey-filling agents (indicated by many, many nonsensical inputs to my sign-up forms). These results thus prompt the next phase of study, in which I take a more exploratory approach to investigate what experimental parameters might elicit more information about each condition's efficacy.

8.5 Phase 2 Empirical Testing

The goal of Phase 2 was to explore more qualitatively how each condition from Phase 1 affects learning and engagement in order to better formulate the next experiment. In this phase, playing happened "live," i.e., in a virtual conference with the researcher, in order to simulate engagement — that is, because the participant is playing while the researcher observes, there is a social encouragement motivating attention, as opposed to a purely online study in which the greatest incentive is remuneration.

8.5.1 Methods

Participants were recruited as described in the iterative playtesting phase (Section 8.3.1) and similarly attended one-hour virtual sessions. However, unlike the iterative playtesting — which was solely the `NT` condition — participants in Phase 2 were assigned to a variety of conditions, as explained in the results. A total of four participants attended six sessions in Phase 2 (two participants attended a second session). Minimal assistance was provided by the researcher, for example to help the participants recover from bugs, clarify the experimental protocol, or answer questions which could be answered by asking other *Foldit* players using the in-game chat.

8.5.2 Results

The first participant (P1) was given 40 minutes to play the `NT` condition, followed by 20 minutes to solve the science puzzle. The participant completed seven levels. The amount of time each level took them supported the hypothesis that a major issue in the design of Phase 1 was that participants were not getting far enough into the game to be prepared for the science puzzle.[9]

In a second session, P1 was given the full hour to continue playing the `NT` levels at a maximum of 5 minutes per level: if they spent 5 minutes on a level, they were asked to move onto the next one. In this way, they saw an additional 13 levels, though they completed only 5 of those. This result, in combination with the other sessions below, indicated that the scope of the task was too large — one hour was not enough time to prepare a player for the science puzzle task.

P2 was given 40 minutes to learn from the `VI` condition — a 19-minute instructional video which suggests, at a couple of points in the instruction, to try playing *Foldit* for oneself to practice the concepts introduced by the video. P2 was allowed to explore the `C` version of *Foldit* in this way. After 40 minutes, P2 had reviewed only 9 minutes of the video, spending most of their time exploring 6 levels. The results of this session suggest that, without further prompting, players easily focus on the game when both the game and video are available.

In a second session, P2 was asked to watch the video in its entirely, then allowed to practice in a sandbox puzzle. In this way, they spent 20 minutes watching the video and 20 minutes exploring the sandbox and various tools (to little success). For the last 20 minutes, they were asked to create the example protein shown at 14:00 in the video by following the exact steps provided between 12:00–14:00. P2 struggled to follow these instructions and was unable to follow along with the video guide.

P3 was given 40 minutes to play the `C` condition. Due to technical interruptions (*Foldit* kept crashing), the session was cut short after about half an hour. However, it was already clear that P3 was on a similar trajectory to P1: they would complete a few levels and be unprepared for solving the science puzzle if given only 40 minutes.

[9]Note that here and in the other sessions, conclusions are being drawn from a sample of only one participant; I want to emphasize that this work is exploratory and not meant to generalize in any way; rather, the conclusions are meant to guide further exploration.

Similar to P2, P4 was asked to spend 20 minutes watching the `VI` video and then given 40 minutes to create the protein shown at 14:00. They were encouraged to ask questions which the researcher would answer — this was designed to simulate a social dialogue or more thorough on-demand assistance from the game. While P4 was better able to follow the video guide, they still struggled with using the tools and operating the user interface. Moreover, they rarely asked questions even when prompted, potentially due to the social pressure players feel regarding low confidence in their own knowledge and a fear of presenting ignorance (cf. Chapters 4 and 5).

8.5.3 Discussion

Although the `CA` condition was not tested,[10] from only these six sessions I already saw a clear trend: the scope of the task I assigned in Phase 1 was too large for participants to meaningfully complete. Participants were completing only a few levels, and often getting stuck, and thus being unprepared when asked to design a novel protein. Therefore, the final phase of empirical testing reduced the scope of what participants were asked to learn and also provided additional learning support, as described in the next section.

8.6 Phase 3 Empirical Testing

The purpose of the final phase was to revisit the testing in Phase 1 but with a reduced scope and more scaffolding to avoid floor effects (i.e., the inability to see effects because differences between conditions are below the minimum value measured). Specifically, rather than try to teach both the prediction and design tasks of *Foldit* (cf. Appendix G.1), this phase focused only on the prediction task. Moreover, in combining elements from the first two phases, participation included a video conference (to prompt engagement) but also included time to learn on one's own, as described in the next section.

8.6.1 Methods

Participants were recruited as described in the iterative playtesting phase (Section 8.3.1). In order to ensure that condition assignment would remain approximately equal during a period of rolling recruitment over several months, participants were randomly assigned to a condition with the least current sign-ups.[11] After signing up, participants were asked to follow a set of instructions

[10]Participants were recruited but failed to attend their sessions.
[11]Because some participants did not complete participation, there was still small variation in sample sizes.

which guided them through downloading a version of *Foldit*, watching a video, and recording their gameplay as they played through some of the game. Afterward, they were asked to upload their recording and fill out a brief questionnaire which included the interest/enjoyment and perceived competence subscales of the Intrinsic Motivation Inventory (IMI) [409, 321]. The IMI is a validated measure of intrinsic motivation, grounded in SDT and often used in games research [409, 527].

Participants were then invited to a 30-minute video conference. During this live session, they were first asked how long ago they played *Foldit* (in days; when a participant mentioned weeks, each week was treated as 7 days). Then they were presented with a quiz about *Foldit* to test their understanding, similar to in Phase 1 (see Appendix I). Like in Phase 1, these test questions were verified with an expert *Foldit* player. During the administration of the quiz, participants were prompted for more detail if needed. For example, if they said "bring the sheets together," the researcher asked what tool they would use to do so.

Lastly, participants were asked to play the Educational puzzle *Mason-Pfizer Monkey Virus*. This puzzle was chosen as a challenging and famous prediction problem with a clear scientific value and no additional knowledge needed (that is, it introduces no new mechanics if one understands prediction puzzles). The participants were advised that because the puzzle is challenging, they were asked only to get the highest score that they could achieve. While they played, the researcher took notes of their strategies and tool usage. A tool was counted as used if it was applied strategically, which was operationally defined as whether the researcher could perceive the tool usage to possibly benefit their fold. This excluded, for example, opening and then immediately closing a tool window, using a tool and then undoing, or using a tool in a seemingly random manner with no perceptible value. When the participant decided they were finished, or after 30 minutes, their final score was recorded. Note that this was not necessarily their best score — sometimes a participant's best score was higher by a few hundred points — however, because there exists a tool to revert to one's highest score and the participants were instructed to finish on their highest score, their final solution was considered a reflection of their ability to use score-manipulation tools and follow instructions.

In all conditions, participants were asked to watch a video alongside playing *Foldit* levels until they completed the target level or could not progress further. For all conditions, the target level was the 16^{th} level.[12] The conditions can be considered a 2x2 design paradigm as follows:

[12]Though, for the `NT` conditions, the last level was described in-game as a challenge and completion was considered optional; therefore, players who reached this level and attempted it were considered to have completed it. Additionally, one player in the `NT+CA` condition recorded themselves completing an additional level and attempting the next one, beyond the requirements of the study.

- `NT+CA` — Given the `CA` video as a guide, follow along in the `NT` levels until the level *Prediction: Collagen* (27:15 in the video).
- `NT+VI` — Watch the `VI` video until the end of the Prediction chapter (11:10), then play the `NT` levels until the level *Prediction: Collagen*, referring back to the video as a guide.
- `C+CA` — Given a similar cognitive apprenticeship video to the `CA` condition, but for the `C` levels and created by a veteran *Foldit* player prior to this experiment,[13] follow along in the `C` levels until the level *The Right Rotation*.
- `C+VI` — Watch the `VI` video until the end of the Prediction chapter (11:10), then play the `C` levels until the level *The Right Rotation*, referring back to the video as a guide.

As in Phase 1, recruitment attracted participants who were solely interested in the financial compensation. I note this because one participant's submission was discovered during analysis to be an exact copy of another participant (i.e., they uploaded the same video under a different name, despite being assigned a different condition). For this case, the second upload was discarded and only the original participation was considered.

After this removal, the final dataset included 28 participants (7 `NT+CA`, 8 `NT+VI`, 7 `C+CA`, 6 `C+VI`). Three participants (1 `NT+CA`,[14] 2 `C+VI`) failed to upload their recordings and fill out the questionnaire; however, these participants were included for analyses on data that were available for them. Because of the expected noise and small sample size, no statistical tests were performed on the resulting quantitative measures.

8.6.2 Results

All conditions were approximately equal with respect to quiz scores, recording lengths, levels completed, interest/enjoyment, and perceived competence (see Table 8.2). Moreover, as shown in Table 8.3, participants were approximately equal in performance on the challenge puzzle. However, the tools they used to solve this puzzle reflect which tools were taught to them. For example, the `NT+CA` condition did not use Pull, the first tool taught in the `C` tutorial, and only the `NT` conditions used Assign SS, Idealize, Ideal SS, Bond Importance, Band Strength / Length, Cut, and Move.

[13] `https://www.youtube.com/watch?v=bIc-FoHLtjY`
[14] This participant reported their play time and levels completed.

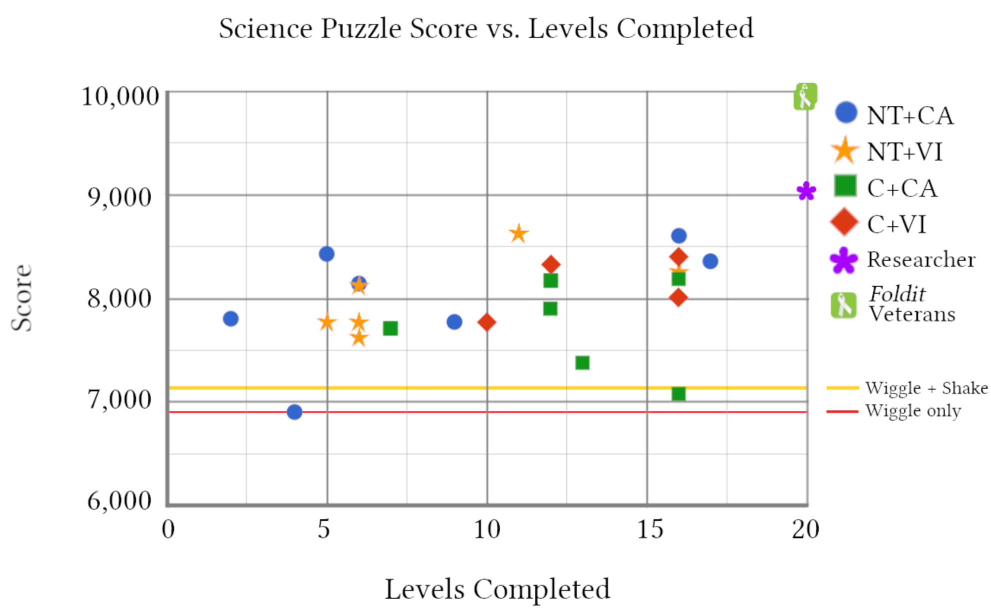


Figure 8.9: Scatter plot of Phase 3 participants' performance on the science puzzle. To contextualize the scores, the red and yellow lines indicate a score achievable simply by pressing Wiggle or Wiggle then Shake, respectively. Additionally, expert scores are given as context: when the researcher attempted this puzzle, he earned a score of 9040. Two *Foldit* veteran players also played this puzzle for their own enjoyment and consented to having their results published: Boots McGraw earned 9926 points; cjddig earned 9970 points. This visualization shows that the highest scores earned were in the `NT` conditions: 8611 in `NT+CA` and 8635 in `NT+VI`. Moreover, there is a slight correlation between levels completed and score. Outliers were excluded from this figure: 0 in `C+VI` (7 levels) and 4532 in `NT+VI` (16 levels). This graph excludes two participants from `C+VI` (scores 7600 and 7853) whose completed levels were unknown because they did not submit a recording of their gameplay.

Condition	Days Since Play	Quiz Score	Recording (mm:ss)†	Levels Completed†	IMI: Enjoyment‡	IMI: Competence‡
`NT+CA`[a]	2.71 (2.56)	2.86 (2.46)	43:26 (12:33)	8.43 (5.91)	4.69 (0.81)	4.17 (1.33)
`NT+VI`[b]	2.63 (1.85)	3.00 (1.69)	37:54 (21:33)	9.00 (4.69)	4.29 (0.56)	3.75 (0.96)
`C+CA`[c]	4.17 (4.92)	2.92 (1.82)	39:14 (13:43)	12.67 (3.33)	3.88 (0.76)	3.36 (0.99)
`C+VI`[d]	3.00 (2.77)	2.86 (1.07)	33:30 (9:50)	12.20 (3.90)	4.91 (1.28)	3.93 (0.93)

[a]n=7 or ‡n=6; [b]n=8; [c]n=6; [d]n=7 or $^{\dagger/\ddagger}$n=5

Table 8.2: Descriptive statistics for each condition. Each cell lists mean (standard deviation). All conditions were approximately equal in measures of engagement and performance.

Condition	Final Score	Time (mm)	Tool Count	Tools Used
`NT+CA`[a]	8007.9 (576.6)	6.0 (2.8)	3.4 (1.8)	Wiggle, Shake, Bands, CI, Assign SS, Idealize, Ideal SS, Tweak
`NT+VI`[b]	7567.1 (1270.6)	8.8 (9.0)	3.5 (1.4)	Wiggle, Shake, Bands, Pull, Cut, Move, Bond Importance, Wiggle Power, Idealize, Band Strength / Length
`C+CA`[c]	7744.7 (444.0)	8.0 (4.4)	4.0 (2.1)	Wiggle, Shake, Bands, Pull, CI, Tweak, Restore Best, Wiggle Power, Tweak
`C+VI`[d]	6854.4 (3036.4)	6.7 (6.9)	3.3 (1.4)	Wiggle, Shake, Bands, Pull, CI, View Options, Freeze, Tweak

[a]n=7; [b]n=8; [c]n=6; [d]n=7

Table 8.3: Participant performance on the science puzzle, *Mason-Pfizer Monkey Virus*. This table shows participants' final scores on the challenge puzzle, the time they spent on it, how many tools they used in solving the puzzle, and a unique list of tools used by participants in that condition. Each numerical cell lists mean (standard deviation). Participants spent approximately the same amount of effort on solving the puzzles, but the tools used reflect which tools were taught in the respective tutorials. `NT+CA` had the highest mean score on the challenge puzzle, though notable outliers include a 0 in `C+VI` and 4532 in `NT+VI`, contributing to their large standard deviations.

Condition	Quiz Score	Tool Count	Puzzle Score	Levels Completed
NT+CA	2	4	8365	17
NT+CA	7	7	8611	16
NT+CA	4.5	2	8432	5
NT+CA	4	4	8151	6
NT+VI	1	6	8635	11
NT+VI	5	5	8270	16
NT+VI	3	3	8128	6
C+CA	2	2	8193	16
C+CA	0	6	8180	12
C+VI	2	6	8016	16
C+VI	3.5	4	8407	16
C+VI	2	2	8328	12

Table 8.4: Performance results for participants with a score of at least 8000 on the science puzzle. This table shows participants' quiz scores, how many tools they used in solving the science puzzle, their final score on the science puzzle, and how many levels they completed. Of this subset, there were 7 participants in NT conditions compared to only 5 in C conditions, and 6 participants in both CA and VI conditions. Participants in the NT conditions overall seemed to score higher on the quiz and earned the three highest puzzle scores of all participants.

When the performances on the science puzzle are graphed, a visualization shows that the NT conditions had the highest scores (see Figure 8.9), only a few hundred points away from what I myself was capable of achieving on the same puzzle. This graph also shows a slight correlation between levels completed and score (though, again, no statistical tests were performed given the small sample size). One noteworthy outlier, though is, a score of 4532 from NT+VI — despite completing 16 levels, this participant submitted one of the lowest scores. This participant, rather than rely on Wiggle and Shake, primarily used Rubber Bands and Cut and Move, focusing on the overall shape of the protein (a strategy used by *Foldit* veterans). So, despite their low score, this participant demonstrated (more so than other participants) early signs of expertise. Indeed, if their solution was given a simple Wiggle and Shake, their score likely would fare with the other highest scores in the cluster of those achieved by participants who completed 16 levels.

Notably, when examining only participants with good scores on the science puzzle (defined by the arbitrary cut-off of 8,000 points, see Table 8.4), we can see that scores on the science puzzle correspond at least partially to expertise, operationally defined as a combination of quiz scores, tool usage, and levels completed. That is, with few exceptions, participants who scored at least 8000 completed most of the levels, performed at least average on the quiz, and used a variety of tools. We can also observe that, of this subset, participants in the `NT` conditions tended to score higher on the quiz than participants in the `C` conditions, suggesting that `NT` participants have a better theoretical grounding of what they are supposed to do, regardless of whether they can practically apply this knowledge (though, given that the three highest scores on the science puzzle come from `NT` conditions, perhaps they can apply this knowledge as well).

8.6.3 Discussion

Phase 3 was a success in that no floor effects were seen. It was less of a success in that little conclusive evidence marked clear distinctions between conditions. We can make high-level observations, such as `CA` conditions being the only data points near baseline performance (Wiggle and Shake), or that after 12 levels completed, the `NT` conditions slightly outperform `C` conditions. Observing their gameplay, I also noticed a tendency for `NT+CA` participants to be more competent in their control over the camera and their "professional vision" (cf. [395]) for seeing the protein as a whole — some `NT+CA` participants would review the entire protein before starting, and notably made use of hotkey shortcuts. One `NT+CA` participant commented after their participation that they would have liked to see more qualitative feedback on their performance, while another commented that the `CA` video was more helpful during the second half of learning (i.e., after they had some practice). This anecdote supports the major finding of Section 8.2.7 — that learning is a cycle of exploring and then learning via social interactions or paratexts. In this case, the participant explored and then used the `CA` video to scaffold their understanding, grounded in their lived experiences.

In reviewing their gameplay recordings, I further observed the patterns of play which cause the breakdowns or breakthroughs [227, 228] of their understanding. A common trend across participants is that they would get stuck on a level for 15–20 minutes, after which they would give up. For these cases, if the game better encouraged them to restart the level, they may have been able to progress further. Part of the issue may have been the stochasticity and "finickyness" of levels — for example, one `NT+CA` participant was stuck on the *Wiggle* level despite following the instructions of the video exactly. Because *Foldit*'s tools rely on complicated, stochastic algorithms, even small changes to the context of tool usage can change the result, making it difficult both for level designers to create playable levels and for players to play those levels.

Additionally, issues which were not raised during playtesting created major barriers for participants during testing. For example, in the level *Clashing Importance*, the level begins in "Exploration mode" (cf. [335]) which allows the player to explore the protein before resetting and continuing with the lesson. Exploration mode is designed to be exited by a trigger through the tutorial, i.e., after clicking through the tutorial instructions. However, for some participants with smaller screen resolutions than tested, the explanatory image was as large as the entire *Foldit* window. Because of this, they responded by closing the tutorial text and unknowingly prevented exiting Exploration mode, resulting in an unwinnable puzzle.

Thus, despite some evidence supporting the effects of the `NT` conditions on engagement and performance, there is not enough evidence to draw significant conclusions. And this finding raises a new issue for game designers — given how noisy realistic populations can be, it can be difficult to demonstrate efficacy in practical settings. These results are unlikely to convince the *Foldit* development team to spend effort implementing features from the `NT` condition because even after 64 participants were tested across three phases of empirical study, we cannot conclusively say that either `NT`, `CA`, or `VI` interventions had a significant effect on engagement or performance. The clearest evidence that the experimental conditions affected performance is shown in Table 8.4, which describes that high-performing participants tended to complete more levels and have a better understanding of *Foldit*'s mechanics (as demonstrated through the quiz and tool usage), and this subset of participants leaned slightly toward `NT` conditions.

How do these results compare to our initial experiment in Chapter 3? In that study, participants completed approximately 9 levels, or a mean of 10.15 in the best condition. The comparable conditions in this study (`C` conditions) averaged over 12 levels completed, suggesting that the improvements were made to the `C` conditions. This could be from the `CA` and `VI` videos, or from other feature changes which were implemented to *Foldit* in the time since the original experiment, or from the face-to-face conditions of this phase of testing (compared to surveying all players of *Foldit*). It is less useful to compare the `NT` conditions because the levels are drastically different in their instructional techniques. (For example, hypothetically, we could make *Foldit* levels trivially easy and inflate this metric, but that would not positively reflect on engagement and performance.)

8.6.3.1 Limitations and Future Work

There are a few explanatory factors for why this experiment had mixed results. First, as described in Chapter 4, there are major limitations of resources (time and money) as well as technical limitations (technical debt) which practically impede the creation of new features. These limitations reduced the scope of the design, meaning that I created fewer practice levels than the theory and best practices would suggest are needed for mastering the skills taught. In fact, players commented on this, asking for more intermediate levels to practice basic skills between learning a new mechanic and applying it intelligently.

Next, any features I added to the game itself needed to be discussed with the development team as a whole. To their credit, the development team was thankfully very open to my experimentation, but it must be said that coordinating with other developers inherently slows the process. For example, as part of pitching the new tutorial, I created descriptions of personas to reflect the use-cases for four different user types. I documented code and coordinated with others on the release schedule of features. While this additional work is useful for the longevity of the game, it slows down the research itself — a balance to be struck and one which presents an inherent challenge for CSGs. Additionally, there were minor limitations with respect to how the quiz was written and graded. For example, many players misunderstood the image in question 4 (which shows amino acids as orange hydrophobics or blue hydrophilics), assuming that the orange segments were actually red and an indication of a low score, to which they suggested wiggling to remove an assumed clash or void. This misinterpretation is likely because the tutorial rarely uses the hydrophobicity view, instead focusing on a score-colored view to orient the players toward maximizing their score (the negative effects of which are discussed below). Furthermore, grading the open-ended answers was subjective, so this aspect of the study would have benefited from additional coders. However, because the sample size was small and the quiz is only an approximation of understanding *Foldit*, the approach I took seemed sufficient for this exploratory study.

There are plenty of opportunities for future work to extend this research. For example, other features can be implemented and measured for their effectiveness (yet, cf. the reflections of the next section). One avenue I would have liked to pursue but did not have time is running workshops with other developers to see if it would be feasible to teach other CSG developers about the insights gained from this work. To what extent can this knowledge be transferred and implemented by other teams?

Finally, considering this entire thread of empirical research on improving *Foldit*'s tutorials, what can we take away as contributions to future development and research?

8.6.3.2 Reflections and Takeaways

Although we might be disappointed about not seeing clear effects from our experimental conditions, this study was never about a binary "were our changes successful or not"? Rather, it was about measuring the practicality of the design suggestions and theoretical insights accumulated across this work. Which features contributed to success? Which had the most impact for the least cost? These results are more difficult to quantify, not just because they target higher-order constructs, but because in-game metrics are not perfectly accurate.

That is, there is only a moderate correlation between *Foldit* score and expertise. *Foldit* veterans admit that the score does not provide utility during the early- to mid-game [332], and previous research on *Foldit* notes that score optimizations do little to encourage intelligent strategic moves [393]. In this way, high-scoring proteins are not always high-value to scientists. If score doesn't capture how useful a protein is scientifically, how else can we measure expertise?

In this study, I operationalized tool usage as tool understanding and quiz scores as mechanical understanding, but these operational definitions may not be fully accurate. Even behavioral measures — observing the strategies players apply — don't represent actually being able to produce useful solutions. Ultimately, the outcome of this study suggests that the metrics used to assess expertise were inconclusive because expertise is not measurable with the common and accessible metrics normally used to measure game performance.

Even early signs of expertise are difficult to create and observe within one or two hours of experience. This was also noted by Ponti et al. [395], who describe how the second author played about 50 hours of *Foldit* just to acquire a basic understanding, recognizing that it would take much longer to function as an intermediate player.

Therefore, I end on a new hypothesis: the way to identify problems with onboarding design in ECCSGs is not through these laboratorial, empirical methods, but through iterative playtesting and longitudinal studies with the community. *That* is my takeaway from this study. If ECCSG developers and researchers want to measure and improve the onboarding experience, it's infeasible to do so with common GUR methods as I've done. Instead, perhaps, a better approach would be to constantly iterate (cf. [564]) and measure longitudinally. Or, better still, start from a simpler concept to begin with, manage technical debt, and avoid feature bloat. Following this advice, iteration and testing can happen easier and quicker because there would be fewer mechanics required to learn in order to play the game competently.

So, anecdotally, what were the cheapest implementable changes? What were the best value-to-investment features? First, **changing the text** was quick and useful. If players were confused by a phrasing, I could easily rephrase it. Similarly, **adding explanatory images** was quick to produce and helped a lot in explaining concepts that were difficult to explain via text alone. Third, adding **dynamic advice** was relatively simple and effective: if there existed a simple trigger mechanism for detecting when a player needed advice (e.g., taking 5 minutes on a 2-minute puzzle or selecting a green segment when they were asked to select a red segment), I was able to trigger just-in-time, context-sensitive help to guide them or correct a misunderstanding. Fourth, **explaining the big picture first** was critical for setting clear expectations and orienting the player toward the core gameplay loop; this was achieved via text and images and, though not captured significantly by empirical metrics, made a big difference to reducing players' confusion. Lastly, although the feasibility of this feature will vary by technical implementation, **signaling** was very useful for onboarding. Directing the player's attention toward the correct element(s) goes a long way in reducing their cognitive and perceptual load and saving those resources for germane processing [500, 192].

What about the opposite — features which did not work as intended? First, the on-demand help menu provided little value; players complained that it was too much information for the tutorial. Second, players complained about the feedback which currently exists for their actions — while the game notes whether one is forming bonds, clearing clashes, and other short-term details, there is no direction toward whether the short-term action is providing long-term benefit toward solving the puzzle. In this regard, the quantitative feedback was merely a reminder to them that the game is not providing better, qualitative feedback. Players wanted more transparency in what the game is measuring, what their actions are doing, and how to achieve their goals.

What features might have been important but did not fit in this study? First, most games build on a cultural foundation of gaming literacy, using standards and conventions for common input schemes and interaction metaphors (cf. Chapter 6). *Foldit* relies on very little of these cultural markers. Similarly, most games rely on some form of social onboarding, such as Twitch streams, YouTube Let's Plays, reviews, walkthroughs, wikis, and other paratexts [94]. But *Foldit*, like other CSGs, severely lacks the quality and quantity of community content needed for a game of this caliber (cf. Chapter 5). Through community content, game learning happens in a dialogue between the player and the community (and its paratexts). And, as a dialogue, players initially absorb from instructors before gradually taking control for themselves (cf. dialogue education [545]). In Chapter 6, I described this as andragogy, a loose hand-holding with available guidance designed for intermediate learners [46]. This is the approach through which most commercial games teach, engaging the players in a "big D" Discourse [181] centered around the game. In trying to change

Foldit's onboarding, I only rewrote the monologue given by the game. I did not — could not — change the dialogue, the Discourse, around it. Yet, the Discourse is where much, perhaps most, of the onboarding value comes from for games with expertise. Tutorials in commercial games — even ones as complicated as *Kerbal Space Program* [485] — typically cover only the basic mechanics (Chapter 6). The deep strategic learning comes through prolonged experience, self-discovery, and social learning.

As of early 2023, *Kerbal Space Program* has 219 "gameplay basics" guides and 154 "walkthroughs" guides on Steam alone.[15] The video "Kerbal Space Program - Career Mode Guide For Beginners - Part 1"[16] by Scott Manley has 1.3 million views. According to VG Insights,[17] *Kerbal Space Program* has sold 4.3 million units to-date — assuming one view per player for the sake of approximation, this would imply that about 30% of the entire *Kerbal Space Program* community has found and watched this one video, out of the dozens if not hundreds of other video guides available online. My point is that expertise-centric games live or die on their community content. *Kerbal Space Program* is clearly thriving while *Foldit*, according to its players who have looked for such help [332], is seriously struggling.

Do these findings transfer to other ECCSGs like *Eyewire* and *Eterna*? I believe they do — as researched in Chapter 5, these ECCSGs share a similar path to expertise, one which relies on learning from the community and practice over time. Changes to the tutorial, and not the discourse or underlying mechanics, can only do so much to smooth out the new player experience.

8.7 Conclusion

In this multi-phased study, I investigated the practicality of implementing various insights gained over the course of this work into *Foldit*. After various empirical tests with a total of 64 participants, there is no conclusive evidence that any particular version of the game or form of instruction was more effective than another at increasing the engagement and improving the performance of its players. On the other hand, this negative result prompts a critical re-examination of the methods used.

The most valuable, easiest-to-implement features were text changes, explanatory images, dynamic advice, explaining the big picture, and signaling. But perhaps more importantly, the diffi-

[15] https://steamcommunity.com/app/220200/guides/
[16] https://www.youtube.com/watch?v=d74m3qThOoU
[17] https://vginsights.com/game/220200

culties in creating a tutorial that generates expertise highlights the value of dialogic, social onboarding. The pre-coded "monologue" of a tutorial, even a dynamic one, seems to have significantly higher cost-to-benefit ratio than having knowledgeable players who can share their expertise with new members of the community. The implications of this research are that developers should pivot away from over-engineering their tutorials and instead grow the community's knowledge and confidence, while making only quick, cheap changes to the tutorial's framing that can (a) recoup big savings for the learner's cognitive, perceptual, and attentional resources, and/or (b) establish earlier and more accurate mental models of the core gameplay.

8.7.1 Takeaways

This chapter was my final attempt to identify and solve problems in onboarding design for *Foldit*. Although I found a few features that cheaply helped — such as explanatory images and guiding visual attention — ultimately, the problem wasn't in the tutorial, it was more fundamental to the game. *Foldit* is too complicated, too noisy, for any programmed instructions to work for every player, especially not when they are given only about an hour and a $15 incentive.

Growing expertise is like growing a garden. It's impractical to measure different growth methods by planting a seed and waiting a week. You have to do the research up front and carefully plan out the future, then constantly monitor the entire bed and adjust accordingly. Plants do not exist in a vacuum, and neither do players.

The community, even veteran players, have not grasped *Foldit* in a way that they can teach and share with others because the game is too complicated to learn on one's own. The cycle of exploring and sharing [332, 240] is stuck because not enough players are reaching expertise to start up the positive cycle of knowledge production. And so, any attempts to create and measure expertise in the span of a few hours will be fraught with noise, false positives, and false negatives. Expertise growth requires a lot of practice and dialogic support, even when the foundational skills are simple and well-taught (though having many or poorly-explained mechanics will make this process much longer).

If there is one takeaway message in my research on improving *Foldit*'s tutorial, it is this: the well-paved road to expertise is a long stretch of practice built by a large community on a few mechanics explained well.

Chapter 9

Conclusion

In Chapter 1, I described ECCSGs as the hardest games in the world to learn. Now, I think they may be some of the hardest games to design for as well. But they have the potential to be deeply valuable — if not critical — to connecting the public to scientific research and enabling truly large-scale scientific efforts.

In Chapter 2, I presented a precise definition for ECCSGs in seven qualitative criteria, and I described the problems best suited for ECCSGs as system-driven, self-contained, complex problem spaces with many instances to solve. This definition is foundational for enabling future work on ECCSGs because it sets the scope for what they can and cannot do well.

In Chapter 3, I started the journey of identifying problems in onboarding design for ECCSGs. Was making a better onboarding experience really as simple as operationalizing and implementing popular theories of learning and motivation like SDT and CLT? No, the features I implemented based on those theories hardly had an impact; but rather, the players' existing axes of expertise mattered. Gamers progressed through the game more; interested biochemists came back for a second session. Players showed differential engagement based on what background they were bringing to the game. This was the first evidence we would see that the player's mental model matters far more than what the tutorial is actually doing — that **the player's background matters**.

In Chapter 4, I zoomed out to get a better sense of the overall pulse of the CSG community. Players reported needs for better scientific communication, instructional design, UI/UX design, and software. Development teams, on the other hand, reported barriers to meeting these needs: limited resources, funding dependencies, internal communication problems, and tensions between scientific research and game design. The problems arising on the player's side of the game can be traced all the way back to how these projects are funded and organized. This research gave the first insights into the **"giant's staircase"** — the problematic learning curve in ECCSGs that is at times too trivial and at other times too steep. It also highlighted the **need for communication**, both between scientists and players and across the player community.

In Chapter 5, I more closely examined the path to expertise in ECCSGs. I found the path to expertise to be a **cycle of exploration followed by social learning**, requiring the motivations of feeling like one is contributing but also gaining something for themself. Yet this path is blocked first by a lack of instruction, then by a lack of polish, and finally a lack of communication. In this way, players need **continued practice** on their own and the ability to **share with the community** so that their personal explorations can crystallize into new understandings and community knowledge. This work also began to highlight the need **showing the big picture**, or explaining up front what mental model the player should have and how the scientific contribution model operates.

Taking Chapters 3–5 together, we start to see a more full picture of the giant's staircase. First, a player comes in with their pre-existing knowledge and interests. They try to understand what the game wants from them, what they earn from contributing, and what they are contributing toward. This onboarding process depends on clear instructional design and scientific communication, but because of systemic issues — a lack of funding and training in teaching and communication — players struggle to understand how to play, how to contribute, and why they should contribute. For players who manage to understand (or press on anyway), they face a new set of barriers: the game has technical issues, poor UI/UX, unclear gameplay, and a lot of mechanics to figure out. These problems are, again, because of the systemic issues with funding and a lack of game design expertise, but also due to the inherent difficulty of integrating scientific concepts with enjoyable gameplay. Meanwhile, because the game is so hard to learn, players aren't confident enough in themselves to share what they know, and their perceived inadequacies prevent them from feeling a sense of belonging with the broader game community, cutting off the production of community content. Community content would have been another avenue for instructional design and scientific communication — even if the team failed to deliver this information, other players could do so through social learning, but social learning has also been ruled out as a consequence of other existing problems. In nearly all ways, CSGs are treated as scientific software, and this mode of operation leads to a focus on the software's capabilities and output — *not*, as suggested in Chapter 3, on the player's mental model and experiences, or as suggested in Chapter 4, on their needs for instructional design and scientific communication.

In Chapter 6, I began looking for solutions. If all of these problems exist, what could the game's onboarding do better? To this, I looked at tutorial design patterns across CSGs, educational games, and commercial entertainment games. Confirming the story so far, CSGs failed to set expectations appropriately (i.e., through scientific communication), and instead introduced mechanic after mechanic without additional practice. They also had less aesthetic polish and more technical bugs and UI/UX issues than commercial games. On the other hand, commercial games demonstrated what success looks like: pacing learning, checking that the player understands, and **keeping mechanics simple**. This work gave us a new tool, a new design pattern, for onboarding design: competence gates — ensuring the player has the skills they need, and if not, helping them get those skills through additional scaffolding. What precipitated from this work was a need for **polishing for playability** — fixing the software issues that create bumps in the learning experience.

By Chapter 7, we've identified the problems in onboarding design for ECCSG. But could we do anything to address those problems? To this, I investigated more closely where the breakdowns are: what skills are players not being taught, and what skills do they need to know? This work further confirmed what we already saw in previous chapters: CSGs need to give the big picture up front to set expectations and communicate the "how" and "why," and players depend on social learning and paratexts when the game doesn't teach them these things. In truth, the other findings from this chapter only serve to support this point: reinforcing the intended structure of knowledge, situating learning within applicable, meaningful contexts, designing for discovery and self-reflection, and encouraging practice and learning beyond the tutorial — these are all about setting the player's mental model, that first insight from Chapter 3, and giving the instructional design and scientific communication — the **explanation** — that the player needs to understand how and why to play.

From the end of Chapter 7 through Chapter 8, I finally tried to implement these insights into *Foldit* to see if we could do anything to address the problems we've identified. Can ECCSG onboarding feasibly be improved, knowing what we know now? To this, I implemented several major features and dozens of minor improvements in *Foldit*'s (ever-aging and increasingly complicated) codebase.[1] And yet, even with these changes, participants shared similar struggles to the previous narrative of this work: mechanics were introduced too quickly, technical issues distracted and confused them, and little more was understood about the science of *Foldit* than from the original version of the game. Why? I believe this result speaks to a methodological problem. Given only an hour or two for the empirical testing I conducted, there wasn't enough time to see emerging expertise. Programmed instructions, even dynamic ones, won't work for every player — that's why

[1] If for nothing else, this dissertation merits a computer science degree for the diagnostic and collaborative efforts it took to effect change in such a multidisciplinary, volatile, and interdependent code environment.

Insight	**Key Elements**	**Guideline**
The Player's Background Matters	Simplicity	Leverage real world analogies, gaming conventions, and other assumptions to simplify how mechanics and dynamics are interpreted.
The Giant's Staircase	Practice, Explanation	Teach gradually through scaffolded whole-task practice.
Need for Communication	Community, Explanation	Communicate the scientific value and progress quickly, simply, and regularly.
Polish for Playability	Simplicity	Address technical bugs and UI/UX issues, and playtest to identify and correct frictions in the player experience.
Cycle of Exploration and Social Learning	Practice, Community	Enable and encourage peer learning among players.
Show the Big Picture	Explanation	Thoroughly, clearly, and quickly communicate the core gameplay loops and scientific contribution model.

Table 9.1: Summary of the insights gained from this research framed as actionable guidelines for CSG development.

social learning, why **community**, is so vital for expertise-building.

What are the commonalities across all of these insights? From the *giant's staircase* and the *cycle of exploration and social learning*, we see a need for **practice** — players need time to sit with the material, explore, digest, and integrate their understanding. This is, of course, seated well within learning science [250, 147]. From the other, *social half of the cycle of expertise* and the *importance of communication*, we see that **community** is also critical to understanding. Players rely on each other and the paratexts they develop, not only as a form of direct, informal learning, but also as a means of engaging with the community, integrating the game into their daily habits, and honing their skills through cognitive apprenticeship. But as Chapter 6 showed, successful onboarding requires some amount of **simplicity** — an overwhelming experience is likely to shy away new players before expertise can develop, not only in how the game is played but in how they perceive these critical paratexts and community content. In this way, simplicity recalls that the *player's background matters*: their prior knowledge and expectations shape how and how readily they perceive new ideas or mechanics. Factor in also that any lack of *polish to the playability* will create major cognitive bumps in the learning process — the game needs some amount of simplicity just to minimize the possibility of technical, UI, and UX issues. Finally, simplicity must be scaffolded with clear **explanation**, including *showing the big picture up front*, smoothing out the *giant's staircase*, and thoroughly, clearly, quickly *communicating* the gameplay and scientific contribution model. Based on these insights and the original observations they came from, I provide a summary of actionable guidelines in Table 9.1.

The other day, I had the opportunity to watch a person play an open-world game for the first time. Despite the game narratively voicing her goal location, signaling a goal marker on screen with a flash, and keeping the goal marker always in sight on the screen, she struggled to identify where her goal was. Even with someone sitting next to her, describing how to identify the goal marker and maneuver the camera to find it, it took some time for her to orient her character toward her goal. I want you to take this example and abstract it: there are some aspects of expertise which a tutorial cannot reasonably teach. It comes with practice. It comes with dialogue. And the more there is to learn, the longer this takes.

To summarize the story of this thesis with one final figure, Figure 9.1 shows my current interpretation of the well-paved road to expertise: a long stretch of practice built by a large community on a few mechanics explained very well. These four elements: **practice**, **community**, **simplicity**, and **explanation**, are what I understand to be the keys to expertise in ECCSGs. Condensing the insights across the empirical chapters, these four words epitomize all contributions of

Figure 9.1: The well-paved road to expertise is a long stretch of practice built by a large community on a few mechanics explained well. Illustration by the author with the use of public domain images.

this dissertation.

Revisiting the research questions from Chapter 1, what did we learn? ($RQ1$) How is ECCSG onboarding design and perceived? ($RQ1_A$) Players experience five major challenges: (1) players are seeking more frequent and clearer scientific communication regarding updates on the projects; (2) players are confused about how to play and need better instructions, (3) user interfaces and controls are often unintuitive, (4) data-focused CSGs suffer from poor task quality, causing player frustration, and (5) CSG software suffers from frequent bugs and crashes that should be addressed. ($RQ1_B$) From the perspective of other stakeholders, there are four other major issues as well: (1) roles are ambiguously allocated; (2) development teams have limited resources and funding dependencies; (3) there is a global need for a CSG community; and (4) science–game tensions create frictions during development. ($RQ1_C$) ECCSG onboarding is a cycle of exploration followed by social learning, but is impeded by missing instruction, missing polish, and missing communication. It is designed to introduce the mechanics in a tutorial with text-box explanations, then provide scientific tasks for the players to complete. ($RQ1_D$) Often, this means that ECCSGs have a separated tutorial rather than integrating learning into the main game mode, and they introduce mechanic after mechanic without sufficient practice and without explaining the game's big picture beforehand, such as through the use of narrative devices. ECCSGs also tend to have more technical bugs and

UI/UX issues which create barriers during learning.

Given these answers, how can we move forward? ($RQ2$) What design changes are practically effective in improving ECCSG onboarding? ($RQ2_A$) Given that players and developers see the tutorial as the only onboarding, and that the tutorial is a passive and standard experience, we need a tutorial that comprehensively prepares players for the gameplay, which means giving the big picture up front and explaining all of the mechanics and the scientific contribution model. ($RQ2_B$) Because ECCSG mechanics are complex, a rigorous skill-based cognitive task analysis backed by 4C/ID might suggest building up whole-task practice. ($RQ2_C$) Yet, this approach elicits expertise details too advanced for practically improving the new player experience. Instead, simpler scaffolding with explanatory images, supporting text, and audiovisual signaling can be much more effective at guiding the player toward quickly understanding the basic gameplay mechanics. ($RQ2_D$) However, these changes taken together do not significantly improve engagement and performance in a real ECCSG; instead, a better approach may be longitudinal — playtesting and iterating with a focus on supporting community knowledge development rather than addressing the design and software of the first hour of play.

Community-focused research would also benefit from a different theoretical framing, such as situated learning [16] and sociocultural learning, rather than my theoretical lens of games as constructionist learning environments [133] and constructivist affinity spaces [180, 460]. Although these findings suggest that my methodology was wrong, we did not know this prior to conducting this research, so I stand by my original justification in Chapter 1 to focus on the first, most visible, and most fundamental portion of the player experience.

What is the bottom-line takeaway for CSG developers? What are the frameworks and processes that work best for CSG tutorial design? My recommendation is to use the Tandem Transformational Game Design Framework [516] to guide your iteration process, but be careful not to get lost and narrow-visioned on specific feedback. When accounting for playtesting feedback, it is easy to focus on the smallest changes that are easiest to make, but often the more important changes (especially early on) regard the game's core mechanics and abstractions. Moreover, all of the academic theories that speak to tutorial design — SDT, CLT, 4C/ID, and so forth — can be helpful for gently pushing the design in one direction or another, but avoid mandating design changes in adherence with a particular theory. They are lenses, not guidelines. Finally, listen to your players' needs, but read into their complaints — player feedback is mired in bias from expert players and idiosyncrasies. As the old game design wisdom says, players can identify the problems, but rarely suggest the right solutions.

Reflecting on the wide array of methods used (as described in Chapter 1), what was gained from taking a holistic, and indeed eclectic, approach to this interdisciplinary research? Ultimately, the answer is triangulation. If only one method found, for example, that clear, frequent scientific communication is an important aspect of the player experience, perhaps it could be an artifact of the study. But such communication stood out as critical across a literature review, player reports, developer reports, qualitative content analysis, reflexive thematic analysis, close readings, and more. In this way, we can be sure that the major findings of this work are multiply grounded in independent data samples and methods.

Why was any of this research significant? I spent my PhD identifying where designers can improve in crafting onboarding experiences for ECCSGs for two reasons. Broadly, I've contributed to our understanding of GBL and tutorial design: by investigating the design challenges of difficult-to-teach games, we can learn how to address those challenges and create more effective teaching in games. Better tutorials means more effective serious and educational games, so this work is valuable to the academics and developers studying and making these games. Fundamentally, this work is a small step forward in making better games and improving everything games are used for, such as health, education, military training, civic engagement, and scientific research [206].

But the second reason I focused on ECCSGs in particular is because of their ability to produce scientific data collection and analysis. If we could make players into experts, if we could make better CSGs, we would get more quality and quantity of scientific knowledge production. We could advance human understanding globally and exponentially. That's the dream, anyway.

To what extent do these findings apply to (non-gamified) expertise-centric citizen science projects? I think that the high-level takeaways — practice, community, simplicity, and explanation — maintain across these projects, indeed maybe even across all expertise-centric domains. Many aspects of learning benefit from practice and social learning, though this "insight" is so broad that it becomes nearly unactionable. The closer one zooms into details of this work, though, the less generalizable it becomes. However, for citizen science projects, much of this work is applicable, especially the stakeholders analysis in Chapter 4 and the path to expertise in Chapter 5, extending Jennett et al.'s MLC model [241].

There are, of course, limitations to how much one can read into my findings, though. Overall, much of this work has small sample sizes, reflecting both the population size of ECCSGs and my limited time and funding. I took a primarily HCI-focused approach, and we would likely see different results if these problems were viewed anthropologically or psychosocially, for example. Moreover, there is inherent bias worth noting as a researcher of *Foldit* who is also developing *Foldit*.

Although my role as a developer was purely for the purpose of this research, I cannot say I was fully detached from the game and its community or that I don't want it to succeed.

That all being said, what comes next? Now that I have defined ECCSGs, identified problems in their onboarding design, and discussed the challenges of addressing those problems purely in the first hour of gameplay, how can the field move forward on understanding and improving ECCSGs? As implied at the end of Chapter 8, I think a good next step would be longitudinal studies. What does growing expertise look like if a CSG team begins with the intention of developing expertise and measures growth over the course of iteration? Furthermore, more design-based research can be conducted on each of the takeaways I've described above. How might we practically design for practice, community, simplicity, and explanation? Although Chapter 8 speaks to this to some degree, that study is markedly exploratory in breaching these topics.

Additionally, there are some questions I've left completely unanswered. What other problems can we solve with ECCSGs? How can dialogic education be integrated into the "official" onboarding flow of a game? To that point, are ECCSGs better taught more like a game or like a professional field? Perhaps, in my focus on games, I have left out the possibility that, actually, ECCSGs *are* more like scientific software, and we should teach them the way we teach programming or medical expertise. Indeed, I think this is an exciting but terrifying time for ECCSGs, because there is so much potential value in this model of scientific knowledge production, yet, if these games don't start genuinely engaging their audiences, the public may become disillusioned from them entirely.

Lastly, I would be remiss not to discuss the recent impact of Large Language Models (LLMs) such as OpenAI's GPT on many areas of problem-solving [65, 503]. Already, these models are being used to more efficiently generate code, analyze datasets, and synthesize information [446]. As I write this, today is March 15, 2023, and just yesterday OpenAI has announced that the next iteration of GPT can take images as inputs and produce text outputs [368]. OpenAI's technical report on this model's capabilities suggest that AI models are rapidly approaching human-level performance on most CSG tasks. Indeed, even two years ago, *Foldit* has seen AlphaFold 2 effectively solve most (or possibly all) problems in protein prediction [478]. The prediction task as tested in Chapter 8 is being automated, and data-labeling citizen science projects like *Galaxy Zoo* [415] and *Eyewire* [408] may soon be similarly solved. But how soon will the rest of *Foldit* become obsolete? What will be left of CSGs when these AI models are more widely available?

While artificial intelligence models may be able to sort, categorize, classify, label, or otherwise perform analysis for data-centric projects, we are far from artificial *expertise*. Protein design remains unsolved and could remain so for some time — any problems which require a combination of creativity, formal knowledge, and task-specific schema are not yet within reach of these task-agnostic LLMs. Moreover, AI models cannot produce genuine human input, so citizen psych-science projects whose goal is to study human cognition (such as *Sea Hero Quest* [226] and *Skill Lab: Science Detective* [463]) are still valuable as CSGs. Similarly, as pure software, AI models cannot conduct data gathering, such as location-based photography tasks like *iNaturalist* [233]. Still, analysis-based data-centric citizen science is perhaps the most common form of citizen science project, and with the rise of AI models that may be able to replace crowds of citizen scientists, we may see a sharp decline in these projects over the next ten years.

ECCSGs, however, as I've defined them, are as of yet irreplaceable. Certainly, AI models can supplement expertise-development. Language learning, for example, benefits from having artificial agents to have conversations with, supplementing real dialog with the community. In this regard, AI can scaffold the practice of certain forms of expertise and potentially provide dynamic explanations. But this does not make ECCSGs obsolete. No artificial intelligence to date can develop novel forms of expertise for solving novel problems. If and when they can, academia as an institution would too be at risk of automation.

So until all of academia is obsolete, ECCSGs are a worthy avenue of scientific knowledge production. They can connect the public to difficult scientific problems and solve larger, more complex problems than humanity has ever solved before by leveraging the scale of a global public. To do this, though, we need the design tools to onboard everyday players into skillful scientists. In this work I argued for four such tools: practice, community, simplicity, and explanation. I've tried to explain what I've learned in as simple terms as possible. Now it's your turn, dear reader, to practice these ideas for yourself, and share with the community what you find.

Part IV

Appendix

Appendix A

Games Studied

Note: For games studied via close play in Chapter 6, refer to Appendix F.

A.1 *ARTigo*

ARTigo is an art history game with the aim of designing a semantic search engine for artworks, in which participants are invited to create keywords and tags for artworks presented in different mini-games [23].
`https://www.artigo.org/`

A.2 *Eterna*

Eterna is a 2D puzzle where players edit a sequence of RNA base pairs to match a target structure [287]. Similar to *Foldit*, *Eterna* features structure design, sequence mutation, and programmable scripts that allow custom combinations of actions.
`https://www.eternagame.org/`

A.3 *Eyewire*

Eyewire is a 3D puzzle where players reconstruct neuron models using 2D slices of serial electron microscopy images [265]. Their success is measured based on input from other players and an initial task-assignment seed from a convolutional neural network [289, 315].
`https://www.eyewire.org/`

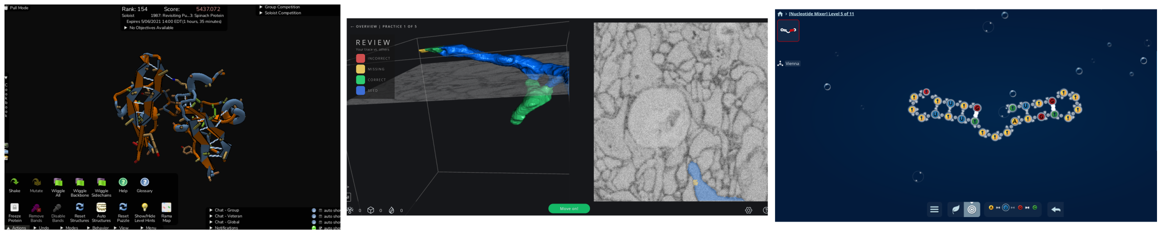


Figure A.1: Screenshots of some games studied: *Foldit* (Left), *Eyewire* (Center), and *Eterna* (Right). Screenshots taken by the author.

A.4 *Foldit*

Foldit [97, 99, 100] is a 3D sandbox puzzle game in which the player attempts to spatially fold a cartoon representation of a protein according to biochemical rules (sometimes clearly represented, but often latent) which affect their overall score. Players manipulate the protein in space using a variety of gameful tools and algorithms such as "wiggle," a local optimizer [99].

This crowdsourcing effort combines human perception and creativity for broad shapes and patterns with computational optimization for refinement to solve two major types of protein-related research questions. First, protein structure prediction: given an amino acid sequence (i.e., the component parts of a protein, also known as its primary structure), predict its secondary and tertiary structure (the shape it will naturally fold into). Second, protein design: given an objective, such as a binding site, design a protein which will optimally achieve this objective. For readers without a biochemistry background, these tasks are analogous to (a) given a set of construction materials, build the most stable building possible, and (b) given a set of architectural constraints (both hard and soft), again construct the most stable building possible, this time while using a (mostly) unconstrained set of building materials.

This second task is similar to the folding task presented in *Eterna* [255, 438], which attempts the same design problem but for RNA. Solving these challenges requires learning complex biochemical rules with limited feedback and knowledge resources, creating the demand for expertise. Also similar to *Eterna*, *Foldit* features structure design, sequence mutation, and programmable scripts that allow custom combinations of actions.

`https://fold.it/`

A.5 *Forgotten Island*

Forgotten Island is a narrativized point-and-click adventure game from the Citizen Sort Project with a mini-game for labeling images of moths [403, 405].
`https://citizensort.org/web.php/forgottenisland`

A.6 *Happy Match*

Happy Match is a game about taxonomically classifying moths, sharks, and rays, from the Citizen Sort project [405].
`https://citizensort.org/web.php/happymatch`

A.7 *Living Links*

Living Links is an online image classification game from the Citizen Sort project about identifying species in Serengeti National Park.
`https://citizensort.org/livinglinks/hmc.php`

A.8 *Mozak*

Mozak is a 3D puzzle about reconstructing neuron models using volumetric neuronal images [435].
`https://www.mozak.science/`

A.9 *Phylo*

Phylo is a 2D puzzle about aligning multiple genetic sequences [254].
`https://phylo.cs.mcgill.ca/`

A.10 *Quantum Moves 2*

Quantum Moves 2 is a puzzle game about solving quantum transfer problems using ludic representations of particles and wave function densities [242].
`https://www.scienceathome.org/games/quantum-moves-2/`

A.11 *Questagame*

Questagame is an image collection game about observing wildlife to catalog biodiversity.
`https://questagame.com/`

A.12 *Reverse the Odds*

Reverse the Odds was a mobile puzzle game based on the classic game *Reversi* with citizen science components for labeling images of cancer slides [481] (no longer available).
`https://www.cancerresearchuk.org/get-involved/citizen-science`

A.13 *Skill Lab: Science Detective*

Skill Lab: Science Detective is citizen psych-science game [239] about assessing cognitive abilities at population-scale, through a suite of psychological tasks as mini-games, for benchmarking and diagnostic purposes [382].
`https://www.scienceathome.org/games/skill-lab-science-detective/`

A.14 *Stall Catchers*

Stall Catchers is an online image classification game about detecting clogged blood vessels in brain images for research on Alzheimer's disease by annotating short video sequences of research data.
`https://stallcatchers.com/`

Appendix B

CSG Player Experience Survey

B.1 Questionnaire

This section describes the full questionnaire given to participants, ordered by relevance to sections in the results.

B.1.1 Participant backgrounds

1. When did you start playing this game?
2. How much education do you have about the topic of the game?
3. What is your level of expertise with this game?
4. How often do you play games?
5. What kind of games do you play? Check all that apply. (See Section B.2.1 for genres.)

For question 1, we excluded 10 invalid answers, i.e., prior to the release of *Foldit*, the earliest one could be playing one of the citizen science games in this report. For question 5, we provided a section for write-in genres. When a write-in answer matched a provided genre, we included it in the count. For example, one participant did not include citizen science but wrote in "Foldit."

B.1.2 Update preferences

We asked participants to rank eight types of updates from 1 (most important) to 8 (least important) — see Figure 4.1 for details on the eight types of updates.

B.1.3 Tutorial experiences

For participants who reported playing the tutorial in the last week (n=104), we asked participants four questions about the tutorial:

1. How difficult was the beginning of the tutorial? (Extremely easy to extremely difficult)
2. How difficult was the end of the tutorial? (Extremely easy to extremely difficult)
3. Of the skills you need to play the game, how many did you learn from the tutorial? (None, some, about half, most, or all)
4. Of the skills taught in the tutorial, how well do you think you could apply these skills to the rest of the game? (Very poorly to very well)

B.1.4 Game difficulty

We gave participants eight statements about game difficulty and asked them to report whether the statement applies for them on none, some, about half, most, or all of the puzzles (or not applicable). The statements were:

1. I feel stuck.
2. The puzzle feels satisfyingly challenging but doable.
3. The puzzle feels too easy.
4. The puzzle feels too hard.
5. I try to look up the answer online.
6. I try to ask others for help.
7. I try to get hints from within the game.
8. The puzzle feels engaging.

Finally, we asked participants to rate the overall difficulty of the game from extremely easy to extremely difficult.

B.1.5 Open-ended game feedback

We asked three open-ended questions:

1. What are your favorite and least favorite aspects of the game?
2. What updates to the game would you like to see the most?
3. What are your favorite and least favorite aspects of the tutorial?

The last question was only shown if the player indicated that they had played the game's tutorial within the last week (n=104; 98 from *Foldit*). This resulted in five open-ended responses per participant, as we coded favorites and least favorites as separate cells. When the response was ambiguous toward favorite or least favorite, coders used their judgment to interpret the participant's intended sentiment, placing the rest of their responses in context. In practice, this was typically enough to reach consensus on a response, but was discussed among coders when insufficient.

B.2 Results

B.2.1 Participant backgrounds

We asked participants when they started playing the game they were reporting on. Their start dates (n=175) ranged from June 2008 to March 2021 with the mean around January 2018. Participant education and game expertise follow a bell curve while gameplay frequency is a bimodal distribution (see Table B.1). The modal participant is a beginner player with novice education (e.g., took a college course on the scientific topic) and plays games daily. Players reported playing puzzle games most (n=103), followed by citizen science (n=99), strategy (n=98), action/adventure (n=83), casual (n=77), role-playing (n=72), and shooter games (n=49). We further analyzed players who reported playing games daily and playing citizen science games as a preferred genre (n=44). Of this subset, participants play puzzle games (n=28), strategy (n=28), action/adventure (n=18), role-playing (n=18), casual (n=16), and shooter games (n=12). From this, we conclude that the modal participant enjoys puzzle and strategy games in addition to their citizen science gaming.

B.2.2 Update preferences

For the remaining closed-ended results (update preferences, tutorial experiences, and game difficulty), because our data is skewed toward *Foldit*, we first sought to test whether we can combine all data for analysis (i.e., to analyze our data as coming from one population of CSG players, rather than two populations of *Foldit* and non-*Foldit* players). To check this, we performed a chi square test of independence on the contingency table of values for the measurements that could be compared (*Foldit*, n=140, non-*Foldit*, n=45). We corrected for multiple testing using the Holm-Sidak method. We found that most of the tests were non-significant, with the exception of responses to the statements "I feel stuck" and "I try to get hints from within the game" (adjusted $p < 0.05$). In

Education	Freq.	Game Expertise	Freq.	Game Freq.	Freq.
None	34	No experience	10	Never	6
Beginner	40	Novice	54	Rarely	32
Novice	56	Beginner	54	Once a month	7
Intermediate	42	Competent	26	Once a week	32
Advanced	8	Proficient	14	At least twice a week	44
Expert	5	Advanced	14	Daily	64
		Expert	13		

Table B.1: Participant reports (n = 185) of education level with the scientific subject of the game, expertise in this game, and how often they play games.

this case, *Foldit* players feel more stuck and seek more hints. However, because most other values were non-significant, we combine all data for the purpose of reporting the remaining results.

As shown in Figure 4.1, players' update preferences are primarily for more scientific news updates. Secondary preferences include more content, new gameplay modes, and developer updates. Bug fixes and quality of life improvements were important to some players but not others. Finally, social and story/gameplay updates were considered least important.

B.2.3 Tutorial experiences

Because our responses on the tutorials were largely skewed toward *Foldit* (n=98), we report only on *Foldit*'s tutorial. As shown in Figure 4.2, the beginning of the tutorial is extremely easy, while the end of the tutorial is moderately difficult. With respect to the skills needed to play, participants reported that the *Foldit* tutorial taught: none (n=1), some (n=13), about half (n=17), most (n=37), and all (n=27). Participants further reported the tutorial taught these skills: very poorly (n=0), poorly (n=8), fairly (n=38), well (n=34), and very well (n=17). From these bell-curve responses, we conclude that players believe the tutorial teaches most of the skills fairly well.

B.2.4 Game difficulty

Participant responses across all games indicated that the puzzles were at a reasonable difficulty. A plurality of 39% of players described that most of the puzzles were satisfyingly challenging but doable, and similar percentages of players said that only some of the puzzles were too easy (50%), too hard (54%), or led to the player feeling stuck (48%). This reasonable difficulty translated well to engagement, as a 41% plurality of players said that most of the puzzles felt engaging. When players were stuck however, they were loath to ask for help — 52% of responses indicated that players didn't ask others for help and 46% of players didn't look up the answers online (for "most of the puzzles"). Players did generally get hints from within the game when stuck though, with a reasonably even spread of answers across the spectrum. A 38% plurality of players found the game "moderately difficult", followed by 23% responding "slightly difficult".

B.2.5 Open-ended game feedback

Using the codebook QCA described in Section 4.1.1, we developed a codebook which ultimately had 23 codes, capturing: educational value, game structure and pace, supporting alternate play modalities, intrinsic game enjoyment (intrinsic game enjoyment), intellectual challenge, socialization and community, boring or repetitive play, gamification, power user functionality and quality-of-life features, user interface and input controls, software, paratexts such as game wikis and YouTube videos, developer communication, scientist communication, making scientific contributions, understanding the science of the game, game difficulty, knowledge of how to play, game instructions (both positive and negative reviews), unknown, and no answer.

To quantitatively analyze the results of the QCA, we summed the counts of codes across coders, thereby weighting agreements more heavily while still including all assigned codes. We report only on the top 1-5 categories for each result; however, the full quantitative analysis is available at `https://osf.io/yd26a/`. For each of the five response types (see Section B.1.5), we explored sums of a variety of subsets of games: *Foldit*, non-*Foldit*, *Foldit*-like (includes *Foldit*, *Eterna*, and *Eyewire*), non-*Foldit*-like, individual games, and all games. We chose these subsets as capturing the diversity of our sample to the extent that we have sufficient data for analysis. However, for this dissertation we report only on findings which showed marked differences between subsets.

B.2.5.1 Favorite aspects of the game

For *Foldit* (n=140): intrinsic game enjoyment (22.7%), educational value (20.2%), and making scientific contributions (17.0%). For non-*Foldit* (n=45): making scientific contributions (23.6%) and intrinsic game enjoyment(17.4%).

B.2.5.2 Least favorite aspects of the game

For *Foldit*: confusion about how to play (19.1%), unintuitive user interface (UI) and control scheme (15.9%), poor quality or quantity of instructions and examples (13.2%), and software issues such as bugs, freezing, and crashes (12.4%). For non-*Foldit*: software issues (16.5%), scientific communication (11.6%), and task quality (9.0%). Notably, scientific communication was highest for *Eterna* (n=14) — which relies heavily on a scientific feedback loop — at 25.0%, and the complaints of task quality were primarily driven by players of *Stall Catchers* (n=14) and *Eyewire* (n=7) — most often regarding data resolution.

B.2.5.3 Updates they would like to see

For all games: power user functionality / quality-of-life features (19.2%). For *Foldit*: UI and control scheme (13.6%), better instructions with more examples and other learning assistance (10.9%). For non-*Foldit*: scientific communication (16.7%) and software updates (10.3%).

B.2.5.4 Favorite and least favorite aspects of the tutorial

Because the majority of our responses came from *Foldit* (n=84; 5 non-*Foldit*) and Wilcoxon rank sum tests indicated significant differences on the closed-ended questions ($p < 0.0001$), we focus our analysis only on *Foldit*'s tutorial and note this limitation of generality. Their favorite and least favorite aspects were identical: instructions (53.1% favorite; 25.2% least favorite) and pacing and structure (20.9% favorite; 16.0% least favorite).

B.2.6 Overall

For *Foldit*: good and bad instructions (18.1%), understanding (or lack thereof) the science of the game (10.0%), and intrinsic game enjoyment (9.3%). For non-*Foldit*: science communication (10.1%), making scientific contributions (9.3%), and gamification (9.3%). For non-*Foldit*-like (n=24): gamification (15.6%), software issues (12.4%), and task quality (9.7%).

Appendix C

CSG Stakeholders Study Details

C.1 Ethnographic Study Methods

The empirical data collected by my co-author Libuše is based on ethnographic research conducted over two years from the beginning of 2020 until the end of 2021 following a cultural anthropological inductive and constructivist grounded theory approach [72].

Three different CSGs were included as case studies — *ARTigo*, *Foldit*, and *Stall Catchers* — of which one case study (*Stall Catchers*) has been studied in-depth and the two other case studies form comparative examples. To gain a deep understanding of the case studies, method triangulation has been applied including methods like participant observation, code, chat and media analysis as well as semi-structured interviews. Whereas the investigation of the focused case study *Stall Catchers* was based on a co-laborative approach [361], participant observation as participant and semi-structured interviews with different stakeholder groups were conducted for all three case studies. Following an ethnographic and grounded theory approach data collection and analysis did not present independent successive but alternating phases. This study particularly draws on the conducted interviews. For the interview recruitment of the CSG team members, purposive sampling was used. In total, across the three case studies *ARTigo*, *Foldit*, and *Stall Catchers*, 7 developers, 4 project leads, 2 community managers, and 8 scientists had been interviewed following a semi-structured interview approach. The participants of the CSGs were invited to participate in the research project via a collective email or a call in the CSG's forum. In total, 30 semi-structured interviews with participants from the three case studies as well as 12 written interviews have been conducted. It must be noted that because of the open call to participate which required interested participants to send an email or message to Libuše, the research participants do not necessarily represent the overall player base of the CSGs but participants who are particularly engaged and/or who chose to share their perspective. The length of the interviews was between 21 minutes and 2 hours and 16 minutes with an average of around 1 hour. All interviews have been audio recorded and transcribed with the consent of the participants. Interviews were conducted in English, German, and Dutch. Non-English interviews have been translated by Libuše.

Libuše Hannah Vepřek's research was funded by the Deutsche Forschungsgemeinschaft (DFG – German Research Foundation) – 464513114.

C.2 Results

We identified the following main stakeholder groups: participants, project designers and leads, professionally trained scientists, (software) developers, game designers, community managers, educators, and students. Other stakeholder groups such as funders and journalists play important roles for CSGs too, but these groups were considered out of scope for the present analysis since they do not interact directly with the production or consumption of the player experience. The definition and division of the individual stakeholder groups has to a certain extent remained an analytical one as the boundaries between, for example, game designers and developers or project leads and the professionally trained scientists are often blurred in CSGs. However, given that each stakeholder group is associated with different responsibilities and tasks in CSGs, the distinction is fruitful for the analysis to gain insights into the specific needs and challenges associated with certain roles.

In the following section, we present the results of the analysis of the individual stakeholder groups. For each group, we first present a description of the role, then discuss their needs and challenges. See Table C.1 for a summary of our analysis. We will refer to statements from our research participants by an identification number and their role(s) as [C]ommunity manager, [D]eveloper, [E]ducation, [G]ame designer, project [L]ead, [P]articipant, and/or [S]cientist. Some quotes are abridged for readability.

Stakeholder	Description	Needs and Challenges
[P]articipants	Volunteers motivated by science and entertainment	Fear of submitting bad data; need to have contributions celebrated; need communication with developers and scientists; need for ethical clarity of their role in science; need for all voices to be heard (especially newcomers)
Project [L]eads	Managers, usually oversee funding and collaborations	Different communication styles within international teams; difficulty building and maintaining community; challenge of handling all of the different roles
Professional [S]cientists	Trained researchers	Challenge of public-facing communication; additional, unusual responsibilities beyond typical scientist duties; discrepancies in funding models between science and game; different goals between scientists and players; difficulties collaborating from outside the core team
Software [D]evelopers	Software engineers, usually students or part-time employees	High churn rate; limited onboarding; required to fill multiple roles; accumulative tech debt; volunteerism; bottom-up development; scope creep
[G]ame Designers	Sub-role of developers (not their own position)	See overarching theme Science–Game Tensions in Section 4.2.2.4
[C]ommunity Managers	Liaisons between team and participants; often a sub-role	Dealing with player pushback; mediating difference between player needs and developer needs; dealing with inappropriate player behavior; labor and skillset not valued by team

[E]ducators / Students	Users of CSGs in the classroom	Need to understand the game better (concepts, controls, gameplay loop, contribution model, etc.); student hesitancy to experiment; challenges with technology; educators need control over content, more educational tutorials, and better support for tracking student progress
Other	External funders, commercial companies in the scientific domain, politicians and policymakers	Not included in this study — recommended as future work

Table C.1: Summary of stakeholder groups and our analysis of their individual needs and challenges.

C.2.1 Participants

The participant's role is at the core of every CSG project. Without their voluntary engagement, the purpose of the game could not be met, the team's efforts would be in vain. As one of the team members describes: "They're everything. They're the most important part [. . .] of the project" [C16].

Although there exist various ways of engagement of non-professionally trained scientists into scientific research — ranging from self-initiated citizen science projects to those that are designed and implemented by professionally trained researchers and developers — the CSGs informing this paper are all examples of the latter. This must be considered when discussing the characteristics and challenges of the participants stakeholder group as they may vary from other forms of engagement in scientific knowledge production.

In these projects, participants voluntarily decide to contribute to a specific CSG (the exceptional case of school children and students will be discussed in Section C.2.5) and actively participate by playing the games at their leisure and thereby contributing to scientific research. In some cases, participants also contribute to CSGs in the context of special events at their workplaces, but in this paper we do not further consider this case separately.

The motivations of participants to contribute to citizen science projects in general, according to Land-Zandstra et al., can range from contributing to "real scientific research or to an important cause such as the environment or health" [283], a general interest in the project's research topic, fun, the opportunity to learn something about a specific research field, and social reasons, e.g., to get in contact with people with the same interests [283]. In CSGs, additional motivations can be enjoyment and complex challenges of the game [114]. Engaged participation with a CSG often requires both the motivation of contributing to the science of the game and the entertainment value of the game itself [332]. In some cases, participation in CSGs can also be a way of coping with everyday life when this is, for example, marked by an incurable disease like Alzheimer's disease or by a pandemic like the COVID-19 pandemic [547].

C.2.1.1 Participant Needs and Challenges

Participants face different challenges that mainly arise from the entanglement of science and play in CSGs. In our discussions with participants, some of them expressed concern about submitting bad data that could harm the research. This fear often derives from a lack of knowledge about how individual contributions and the results are calculated. Although out of scope for this study, we refer to previous literature which has investigated how to match participant skills to appropriate tasks and how to celebrate individual contributions [282, 484, 240, 392, 449, 448].

Participants are very focused on their contributions. The moments described as most frustrating for participants often refer to the feeling that their contribution is not valuable. For example, one participant explains that "[i]f you get [bad quality data] too many times, you lose interest because it's, you just fear that your work is meaningless" [P17]. This meaninglessness occurs when the research data analyzed is of bad quality or because technical problems and bugs in the code make it difficult for the participant to contribute in a satisfying and meaningful way.

This challenge is also connected to the next one, which regards communication with the CSG team. Although participants positively mention the possibility to communicate with the team via in-game chats and forums, they express their dissatisfaction with the way developers or the team in general handle bugs and issues reported by players. When asked if they would report bugs to the team, one participant described reporting a bug and seeing the bug still present six months later without a response from the developers. Participants have different understandings of the priorities than the development team:

> *[W]hat I find disappointing is that they are now busy with [some new feature] while they also have bugs which are serious and which actually should have been solved first.*

> *[...] [M]y focus with my software developer background would be: fix the bugs first before introducing new features.* [P18]

This communication problem is likely bound to the lack of information about — and intransparency of — processes within the team and the lack of resources that constrain the developer's work. At the same time, this illustrates the power hierarchies within these kinds of citizen science projects: it is the scientists and developers of CSGs who primarily set research goals and priorities, not the participants.

Lastly, there are two challenges with the role of participants, generally. When developers try to survey the participant community to understand how to better serve them, only the most active players engage or have opinions [C9]. Similar effects have been found with commercial games [372] — ultimately, the voices of new participants need to be heard, but measuring their opinions is challenging.

The second issue with participants, generally, is that they often fall through regulatory cracks due to their ambiguously defined role which relates to our larger finding about ambiguous roles, discussed in Section 4.2.2.1, but we focus here on how it affects participants in particular. This becomes apparent in Institutional Review Board (IRB) processes evaluating CSG projects where the role of participants moves between the categories of "human subjects", "research participants" and "scientists" [436, 427, 546]. In citizen science in general, and therefore also in CSGs, participants can sometimes even be both researchers and human subjects [427]. It is not uncommon in CSGs that some very engaged participants take over additional roles, such as community management, besides their contribution as participants. As moderators, these individuals take on important tasks to maintain the project — for example, they are often the ones who report bugs or problems, but they also act as intermediaries between the team and the participants. While participants volunteer to take over these roles and are publicly recognized for their additional commitment on the project's website or communication platforms, this nevertheless raises questions about the lines drawn between compensated and uncompensated work. Moreover, these different role understandings not only challenge oversight committees — which so far have been particularly focusing on the protection of human subjects [427] — but also the CSG stakeholders investigated in this paper.

Ultimately, the role of CSG participants in the ethics of scientific research remains a necessary conversation with the greater CSG community.

C.2.2 Project Leads

The second stakeholder group is the project leads (or managers / project designers) of CSGs. As project leads often take on many different roles, this stakeholder group is not always distinguishable from other stakeholder groups like scientists or developers, and the lines between the groups can be blurred. We define a project lead to be the role of managing the design, development, and maintenance of the project and the team. Often, the project lead oversees the funding of the project, and when the project lead is also a scientist (which is often the case), they determine the direction(s) of the game's research. Project leads are in charge of the overall direction of the project and formally representing the project, but also go *[...] out for collaborations and connecting with the community* (paraphrased field note from [S8]). In many CSG teams, these decisions are jointly discussed among the team and tasks are divided between different team members.

C.2.2.1 Project Lead Needs and Challenges

In total, we identified the following three recurring challenges described by project leads of citizen science games: 1) different communication styles within international teams, 2) difficulty to build and maintain community, 3) challenge to handle all of the different roles. A fourth outstanding challenge is a lack of resources, which weaves into all other challenges. Because of their huge responsibility for the CSGs, project leads have to deal heavily with acquiring and managing resources. However, as this challenge affects almost all stakeholders involved in CSGs, we will discuss resource issues in Section 4.2.2.2.

The first challenge we identified for the project lead stakeholder group refers to the distributed team structures and mainly remote collaboration. It is not unusual for CSGs to be developed by a team that consists of team members spread around the country or even the world and from different institutions. While these structures may offer some physical flexibility to one's work, they more often present challenges due to different communication styles and availability within the teams.

Not knowing the target audience and the motivations or needs of the users also makes it incredibly hard to build and maintain a community of a CSG [S8], which forms the second challenge for project leads [ELS15]. "Community" here refers to the collection of CSG participants (or players). "Focus on the community, That's what will make or break your project", says [DP13]. Building a community is of utmost importance for CSGs, as these projects are dependent on the ongoing contributions by participants and being part of a community has been described as motivating by participants [513, 229, 114].

At the same time, building a community is no easy endeavor. There does not exist a generic "how to" approach and every community is unique in its shared ideas and what binds them together. It is even questionable if a community can be formed from the outside or if it has to grow from inside.

The third recurring challenge we identified for project leads is the challenge to handle all different roles. As described in the beginning of this section, the role of project leads is not always clearly defined. Because of the mostly small team sizes and lack of resources (with respect to time, funding, and team members), project leads have to step into all of the different roles and tasks. It is rare that one person can adequately replicate the expertise of many roles, which results in one or more of these jobs being insufficiently performed.

C.2.3 Professional Scientists

By "professional scientists," we refer to the professionally-trained researchers who lead the scientific investigation behind a specific CSG. In the case studies examined, they are the ones who define the research questions, the methods, and analysis, and who decide how to include the crowd into conducting the research. Professional scientists also write-up and publish the research results and are involved in funding for the projects. In many but not all CSGs, professional scientists also take over the role as project leads. In this section, we focus on the aspects which especially concern scientific and research tasks within CSGs.

For the interviewed scientists, the purpose of CSGs is both to help science, e.g., by accelerating the analysis of research data, and to connect people to science. Some also stressed the potential of the CSG to educate people, like one scientist who explained this to be the main "mission" of the CSG: "I think that the scientific aspects of it are quite valuable but to me that is definitely secondary in my evaluation of it" [S20].

At the same time, CSGs can help scientists reflect on how to explain and present science to the public. By conducting research in view of the public throughout the process, the scientists get both practice explaining the research as well as feedback on what players understood and how the CSG is effective at assisting this research or not: "Having players involved in development is really, really good... and immensely valuable for the research" [DS5].

C.2.3.1 Scientist Needs and Challenges

In practice, though, scientists face difficulties with the kind of public communications described above. For example, because scientists are so interested in the topic, they sometimes struggle to understand how to motivate and engage educators and students who don't share the same passion [S8].

Moreover, this additional side of research creates more responsibilities for the scientist. Being the lead scientist on a CSG becomes itself a full-time job, inhibiting their personal academic careers [ELS15]. This includes additional responsibilities for marketing and advertising: normally, scientists don't often need to worry about broadly marketing their research, but for a CSG engaging a wide public audience is essential for adequate quantities of unbiased data [ES1, S2].

Challenges also arise from the discrepancy between the current rationale/logic of academic science and CSGs as game platforms and community outreach projects. Today, the success and careers of professional scientists heavily depend on scientific publications. However, as one scientist explains, this is not always possible with CSGs which mostly have to be designed as long-term projects to build a user base:

> *[I]f we can't get any research results or publications out of it, then we can't put any work into it. So that's typically how it is, projects in computer science are always some kind of research prototypes that are implemented to generate some kind of data or to test the validity [...] And then either the PhD ends or funding ends or something like that and then it's either discontinued or it's somehow taken over as a product [...] by some department that takes care of it. So that's the theory. And the second part never really happens.* [DS21]

The described discrepancy is also experienced in the application for funding. Interview partners expressed the problem that mostly only new, innovative research projects would be funded. However, the realization of a CSG — which could then support innovative research — would require funding to implement the platform, building on existing and well-established solutions. We discuss financial modes of operation more in Sections 4.2.2.2 and 4.2.2.4.

Another major challenge is the different understandings and goals between professional scientists and players. In some of the present case studies, the participants' and researchers' aims do not always align because of the game characteristics which would sometimes conflict with the research goals. For example, the "game" would afford [183, 33] and encourage participants to focus on earning points even when more points would not translate into more accurate or interesting scientific results [ELS15]. This tension has also been observed by Ponti and Stankovic [394] for the case of Foldit where certain player behaviors (scripting) produce high-scoring solutions which are not necessarily scientifically valuable [394].

Although developers often emphasized the discrepancies between the scientists' goals and players' goals, it is worth noting here that scientists and developers working on a CSG sometimes also strive for different goals. While scientific accuracy and confidence is most important for scientists, game developers aim to "make [the CSG] bug free" [D22].

Lastly, the issues described so far are primarily concerns for the lead scientist, but what about scientific researchers who are not part of the core development team? There is a need for CSG teams to collaborate with other labs and researchers, yet there exist barriers to those collaborations, as detailed by [ES1]:

> *Whereas right now, in order to do it, you have to know somebody. You have to know somebody who's involved in [game]. Or be willing to send an email cold to someone who's involved in [game], have a conversation about what it is that [...] players can do and how they can help, you know, go through kind of a vetting process, probably go to two or three meetings. And then* and only then *will your science be ready to submit to [game] players through this rather laborious and time consuming and potentially daunting process. Whereas there is no particular reason why we can't make this information more readily available to the science community so that they can actually do more of it on their own. [...] If [game] is going to be a big resource for the research community, it has to have a wider base of players as well as a wider base of researchers.* [ES1]

In summary, being a CSG scientist can be a difficult and full-time job. They are challenged with public-facing communications and education, marketing and recruitment, the academic demand for publications, difficulties funding long-term CSGs, discrepancies between player motivations and scientific goals, and barriers to collaboration with third-party researchers.

C.2.4 Other Team Members

Within a CSG development team, there are generally three subroles. Software developers create the front-end and back-end technologies for the application and website. Game designers design and implement gamification elements. And community managers form the bridge between developers and participants.

C.2.4.1 (Software) Developer Needs and Challenges

Despite being important workhorses of the team, the developer position is rarely a full-time one — most developers have job requirements besides working on CSGs. In fact, the bulk of software development is often done by students with a professor acting as the project lead. Given the lack of team members described above, there are not enough developer positions to encourage specialization, so most team members work across the full stack of software development.

The first issue developers encounter is the onboarding process. When asked what the onboarding was like, one developer said "None. And terrible" [DS5]. There is a need for documenting development protocols to counteract the steep learning curve for developers, yet developers acknowledged their documentation and documentation process was weak or fledgling.

Because developers are often students or part-time, there is also a high churn rate for CSG developers. This means that there's no guarantee a developer will be able to make productive contributions to the project during their time on the team. In fact, new developers can often weigh down the project by requiring a lengthy onboarding (to what can be a very large codebase and assemblage of operations) and leaving shortly after onboarding. Moreover, because of the lack of documentation, when a long-time developer does leave, the knowledge they gained — e.g., about handling specific bugs or codebase quirks — is often taken with them, lost to the rest of the team.

This high churn rate also seeds distrust among the players when a new developer joins the team; the player community questions what their contribution will be:

> *Graduate students come in. They work on a project for a year or two and then they leave. And so. [...] It's really unclear. Like, are you going to be helpful? Do you care about the community at all?* [DS5]

Being short-staffed, developers also need to learn new skills and take on jobs outside of their traditional roles — or what they're even qualified to do. Examples include developers as game designers, community managers, artists, or marketers — or vice versa, wherein these positions also require coding expertise (e.g., community managers interfacing with a SQL database) to perform their normal jobs. This is especially true as developers who are not trained in community management describe great difficulties in building and maintaining a community, as described in later sections.

Perhaps the biggest challenge, however, is a fundamental lack of resources. As noted earlier, developer time is limited due to funding restrictions. This has several downstream effects, including a backlog of bugs and a long list of "we should really do these things" [DP13]. This results in bugs only being fixed when they pass a threshold of player complaints [C9]. Eventually, this accumulates into "tech debt" — the developers must deal with old, poorly written code, inflexible systems, and "patchwork upon patchwork upon patchwork upon patchwork as the project had grown [...into a] complete mangled mess" [DS5].

This is further exacerbated by a lack of funding to address tech debt; however, this issue will be expanded on in Section 4.2.2.2. We summarize two notable effects of the issue of resources: first, developers must often volunteer their time, since there is no budget to employ them for their work. Second, being grant funded, development happens bottom-up rather than top-down: grants fund particular features or datasets, rather than contributing to the holistic design of the game. This lack of overarching vision leads to disparate development movements happening simultaneously. The downstream effect of this is that developers struggle to come to a consensus on the look and feel of the game's design, UI, and onboarding [S8].

Between the bottom-up development culture and developer volunteerism, many developers often end up working on whatever interests them, rather than doing the work that would be most helpful to the project. This easily slips into scope creep as new features get introduced. Yet, these features can easily get abandoned if the one developer spearheading that feature leaves [ELS15]. And this phenomenon is not limited to user-facing tools — software development workflows such as Jira can also get started and abandoned, creating increased difficulty for new developers trying to understand the project as it's spread out sporadically and inconsistently across multiple tracking softwares [ELS15]. In fact, one developer [DGL4] mentioned preferring unfunded work because of how it allows more control over the scope of the project.

Lastly, though not a direct challenge, it's worth noting that there is currently little overlap between CSGs and the gaming industry. CSG developers aren't engaging with the industry, and likewise the industry doesn't recognize citizen science games. Because of this, CSG developers may not be aware of best practices for the design and development of commercial, mass-market games, and professional developers have little to no interest in supporting CSGs — as contrasted with, for example, "indie" development, which veteran developers happily support pro bono (e.g., [418]).

C.2.4.2 Game Designer Needs and Challenges

Game designers are worth noting for one particular feature: they are never, in our data, their own role on the core team. Despite the gaming industry seeing game design as a wholly distinct position from game programmer, the two are not made distinct in CSGs. Yet, where "game designer" exists as a concept, they have their own role-specific challenges.

The challenges of the CSG game designer can be summarized as a tangle of tensions between the game and the science of citizen science game. There are three components to CSG development: the science, the software, and the game. Science and software can coordinate because scientific software is a common practice, both are familiar with operating on grant budgets and deploying feature by feature, test by test. Game and software are similarly in agreement, since video games are inherently software and game designers and programmers alike are familiar with the fast pace of iterative design and development. Yet, when game meets science, this is where practices diverge. The tension between science and gaming will be further unpacked in Section 4.2.2.4.

C.2.4.3 Community Managers Needs and Challenges

Community liaisons, community managers, and other outreach roles form the link between the participants and the developers and scientists behind a CSG. Usually, they monitor all communication media and platforms available for participants to connect with the team and with each other, such as in-game chats and forums. Besides being responsive to the CSG community and forwarding requests and questions from the participants to other team members, community liaisons also translate the needs of the participants for the team: "[T]rying to connect them in understanding, in having the developers know what [. . .] players are really looking for." [C23]. In this way, community liaisons can also be understood as the advocates of the participants in the CSG team. At the same time, they often also communicate in the other direction by taking the science "behind the project and translating that into human language" [C16].

This can also create issues when discussing the realities of science. For example, [ELS15] describes a time when their game partnered with a pharmaceutical company in order to further fund the game, which created vehement pushback and distrust from players. Because the team didn't properly explain the situation or get participant approval, they reflect, the team lost credibility as a non-profit and unbiased third-party working in the name of science. After that incident, they were hesitant to be open about the team's intentions and logistics. "We walk a very fine line between telling them too much and then not telling them enough [...] we tend to do that for the science. 'Oh no, we don't tell them about that. Yet"' [ELS15].

Whether due to pushbacks such as that, or other issues causing lack of trust, community managers are encouraged by the rest of the development team to not be fully transparent about the development process. One community manager [C9] described taking a list of bugs to the developers and being told to say the team was looking into it when it was simply low priority:

> *[W]hy can't you [the developers] take ten, literally ten minutes to look at this thing, for example, or or just all kinds of small little things like that. [...] [W]e had to just, y'know explain it away and go 'oh the developers are working on it.' And [...] they're not.* [C9]

On the other side, community managers also have to deal with inappropriate behavior from players. In some cases, this means not putting effort into the scientific task in order to simply play the game [DGL4]. In other cases, players will look up the answers to tasks, making benchmarking performance difficult [ELS15]. Ultimately, CSG teams need to be prepared for players to cheat and exploit the game, because it will happen and it will add toxicity to the community [C9] (cf. [41]). According to [C9] and [DGL4], player behavior is driven by whatever is incentivized most by the game system, whether that aligns with the scientific goal or not; it is the responsibility of the CSG team to expect this exploitation and minimize it up front, rather than waiting to address those loopholes.

> *You also have to bear in mind that once there's compensation involved, if people get invested [...] cheating* will *happen [...] even though it is a citizen science game, everybody's here to do science together for a higher good. As soon as it's a game, you will get people who want to break the game.* [C9]

To make matters more difficult for community managers and liaisons, there is often no designated community manager role. In most cases, the position is part-time or combined with other responsibilities within the team.

Taking over the communication between the CSG team and participants can sometimes be challenging for community liaisons as they have to be responsive to participants but at the same time they are often dependent on the scientists and developers to answer specific questions. Moreover, what has been described as another challenge by team members in the community liaison role is "juggling all the different roles" [C16] which stems from the fact that it is usually not a standalone position but a role integrated into other tasks.

On top of this, the work done by community managers — public communication, mediation, and emotional labor — isn't valued or acknowledged as a skill set.

> *Our biggest weakness, and this is across the board, everybody involved with [game] at all, was our lack of — this is going to sound stupid — customer service skills. [...The principal investigators] know how to interact with students, postdocs, [etc.] they got*

> *that down. [...] The students [...] know how to interact with other students in an academic setting. But. When it comes to interacting essentially with the general public, we were garbage.* [ELS15]

One participant describes how community managers are treated negatively for relaying negative feedback from the players [C9]: "Asking for bugs to get fixed was wasting developer time or something, like that was the impression that we sort of got [...] We bring up concerns hypothetically, but we know that the answer is always going to be OK but [the developers] need to be working on this thing for [the lab] right now." This phenomenon is confirmed from the other angle by a researcher who acknowledges these flaws: "We're just really bad at addressing anything that isn't like the game is broken and it's the end of the world" [ELS15]. As discussed in Section 4.2.2.4, the players' needs for bug fixes and experience improvements are treated as secondary to the needs of scientific development.

Where community management exists, these team members are the only bridge between the team and the players. For one case study, community management (when considered as its own job) has historically been a feminine role, entangling gender privilege and power dynamics into the interplay between supervisors, developers, and community managers [C9].

Most developers have no direct interactions with the players [S2, S8, PD13]. In one project in particular, players take on a more driving role in the experimental design, collaborating with the scientists to iterate on the research questions and critically examine the experimental process [PD13]. However, the player perspective is being understood here only from the scientific angle — rarely do the teams directly correspond with players regarding their game experiences.

To summarize, community management is a critical and undervalued responsibility on the CSG team. The struggles of dedicated community managers emphasize that player experiences are treated as secondary to the scientific research, while part-time community managers recognize their lack of skill in adequately communicating with the player base and general public.

C.2.5 Educators and Students

Although citizen science games were originally designed for scientific value, and to engage gamers, they have found use in educational settings as well. Several educators have started making use of citizen science games in their classrooms for their value as an interactive learning experience that illustrates concepts with immediate feedback [ES1, E3, E6, E10]. The immediate game feedback is also useful for automated grading, since games come with built-in scoring systems [ES1].

For some, the game aspect of CSGs was also helpful for motivating students [ES1, E7] — as [E7] describes, “When I say game... they perk up” — though other educators didn’t focus on its gameful nature. Similarly, some educators leaned into the citizen science aspect of CSGs [ES1, E3, E10], using the game as a springboard toward curiosity — discussing how science happens in real life, engaging with current events, and connecting students to a larger community of people interested in science — while other educators focused just on the software itself. One participant [E10] shared a story of having her students watch a video blog and having a scientific question. She emailed the scientist with their question and heard back immediately, giving the students an interaction with a real scientist. ”And I thought that was one of the most successful and fun parts of the project,” she said.

C.2.5.1 Educator and Student Needs and Challenges

What are the challenges of CSGs in education? First, educators and students are not understanding the game. Partly, this is because the tutorials are not helpful for them and the game is not well-explained [E7]. Educators also describe how the controls were unintuitive, and it took them a long time to learn how to play — despite their scientific knowledge on the subject, which didn’t help. “I did not get the sense that me thinking through the science was going to help,” said [E10]. This participant also described feeling personally responsible for their failures in the game, spending as much as several hours on a single puzzle, sometimes needing to “cheat” by looking up walkthroughs or video guides.

Moreover, some educators are skipping the tutorials and/or not playing the game themselves, despite assigning the game to their students, which furthers the lack of understanding. In part, this disinterest in playing is due to the game’s design. [E7] commented that there isn’t “a good rewarding system,” explaining that other games have achievable goals and rewards for getting to the next level. For some games the reward is getting a paper published, “but how often [is that] gonna happen when I play?” [E7]. Other times, educators are trying to understand the game but failing or taking a long time to do so. “I’d spend like an hour or two most days, like futzing with the tutorials,” says [E10]. These sentiments combined paint a picture that educators are struggling to understand and get involved with the game, which in turn makes it difficult for them to get students interested or help students understand how to engage.

Turning now to the student experience through the lens of educators, the students’ lack of understanding comes in part from a hesitation to experiment and a “finicky” nature to the tutorials

[E3]. [ES1] said that students have a "hesitancy to try stuff" and a "fear of breaking things." Additionally, other educators mentioned difficulties navigating the user interface and understanding the unintuitive controls [E10]. [E6] notes that this is partly because students come into the classroom "with varying degrees of comfort with technology and varying levels of willingness to experience frustration while they're doing something," a statement echoed by [E10].

And when they are bold enough to experiment, their experiments fail. "The feedback I got from my students is that what frustrated them was when they couldn't get past a level, even after following the directions that were outlined in the wiki and watching other videos of how other people had done it" says [E3]. For [E10], she describes needing to find the "magical combination" of fiddling with the puzzle that solves it. She further notes, to her dismay, that her scientific knowledge as an educator didn't apply much to solve the puzzles, which may be considered a separate issue in integrating education with citizen science games. Even the beginner puzzles are complicated and require trial-and-error, says [E6], "And some of the students find that tedious or frustrating." To resolve some of their frustrations, educators tend to allow students to work in groups or in class with a partner [E3, E6, E10].

Moreover, when given the game during class activities, students ignore the didactics and just play the game as a game. And in doing so, the relative learning gains are inefficient. [E3] comments, "The biggest criticism was that I think because they didn't get a whole lot out of it, that the amount of time that they spent completing the assignment wasn't proportional to what they were getting out of it."

For children (7th–10th grade), one educator describes how the students can focus on a game for "maybe 15 minutes" [EP24]. The characters and competition of a CSG are helpful enough to be a change in the classroom, but not enjoyable enough that students would leave their consoles to play [EP25].

Given these challenges, what are the needs of educators and students using CSGs? First, the educators need more control over the game content, making tutorials either more educational or skippable (or both). If the game teaches only basic concepts, the activity will likely not be valuable enough relative to the time it costs to learn the software [E3].

Second, the educators need better support for tracking student progress. As suggested by [ES1], CSGs can be used for automated grading by their interactive nature, but this requires systems that support tracking student progress relative to the learning material, not just relative to the gameplay. As recommended to educational game designers, games that are used in the classroom should be designed with careful consideration of what data are collected by the game and how those data are presented to educators, e.g., through dashboards of clear data visualizations [270]. Moreover, if educational use is something that the CSG supports, then data collection and visualization should be considered throughout the design process, rather than as an afterthought [270].

Third, to better connect with the citizen science value of the game [ES1, E3, E10], educators would benefit from more detail on what science is happening and how it is integrated into gameplay. Fourth, educators reported that students struggled often with technical issues, including institution-specific problems [E3] and issues downloading and installing the software [E3, E10]. Making CSGs more accessible on the hardware available to students and in the classroom (e.g., in web browsers), would greatly increase potential benefits of connecting citizen science games with formal education.

Finally, the greatest need of these stakeholders — which encompasses all prior needs — is that CSG developers need to be collaborating directly with educators. Currently, CSG development is not seeking to meet the needs of educators, and this results quite expectedly in educators not having their needs met when trying to use citizen science games in the classroom.

C.2.6 Other Groups

There are several other stakeholders involved in CSGs. This includes funders who support development — both organizational funders, such as the members of NIH, NSF, and private companies and investors who donate to or invest in scientific ventures, as well private funders such as philanthropists and other donors. Another group of stakeholders includes members of third-party for-profit companies within the scientific domain who collaborate with CSG teams on specific projects. Relatedly, there are other companies in the supply chain and market of the scientific domains. For example, when *Eterna* synthesizes player-made RNA designs, they must interact with companies who produce the scientific equipment and consumable scientific products used in wet laboratory experiments [287].

For environmental citizen science, residents of the environments in question have inherent interest in project outcomes. Lastly, CSGs are influenced by politicians and other policymakers who set laws and regulations regarding science, software, and game development. As discussed previously, we excluded these groups because they do not interact directly with the production or consumption of the player experience. However, future work should examine how these stakeholders further factor into the greater CSG network.

Appendix D

Code Descriptions

Here I briefly describe the codes generated in the study from Chapter 6; a more detailed description can be found in the codebook at `https://osf.io/ut4mg/`.

- **Aesthetic Polish** — The game has a cohesive aesthetic and well-polished music, art, sound effects, and visual animation.
- **Assume Game Literacy** — The game requires that players have some amount of game literacy.
- **Camera $>$ Controls $>$ Interactions $>$ Mechanics** — Where applicable, first camera controls are taught, then other basic controls like movement, then fundamental interactions, then core gameplay mechanics, and finally ancillary mechanics.
- **Can't Go Back in Tutorial** — During play, the player tried to go back in the tutorial but was unable to.
- **Citizen Science Explanation** — The citizen science game describes its citizen science purpose.
- **Clear Design Language** — The game uses visual and/or audio cues for a clear, specific mechanical meaning, i.e., a mechanical indication to the player.
- **Competence Gates** — The player must demonstrate an understanding of a mechanic before being allowed to continue with the game. This happens multiple times with different mechanics. (e.g., it is insufficient for one test of 'player can control their character and follow the basic core mechanic').

- **Customizable Character(s)** — The player can customize one or more avatars which they control or which represent them.
- **Details Unexplained** — Details of the game mechanics, such as how mechanics interact or what powerups do, are not explained by the game.
- **Dynamic Help** — The game presents additional support when it detects that the player is experiencing difficulty.
- **Educational Value** — The game had learning and/or the expectation of learning, coded as more than expected, less than expected, or as expected.
- **Felt Competent After Play** — After the session, the researcher felt competent in their abilities to succeed at the game, comfortable in the gameplay, educational materials, and/or citizen science components.
- **Forced Exploration** — The game's instructions were insufficient for thorough learning, so the player felt that they had to learn details of the game for themselves.
- **Gestures** — Some of the game's instructions are given nonverbally through animated gestures, such as a hand or cursor showing how to interact.
- **Good Feedback** — Feedback is immediate, clear, and consistent.
- **Gradual Complexity** — The game gradually increases in complexity.
- **Hints** — The game provides on-demand hints for the puzzles it presents.
- **Just-In-Time (JIT) Tutorial** — Information is presented just-in-time for solving the tasks which require that information.
- **Level / Environment Design** — Some of the game's instruction came implicitly through design and/or structure of the (non-tutorial) levels and/or environment.
- **Mechanic After Mechanic** — The game's tutorial introduces new mechanics one after the other without space for practice between the introduction of new ideas.
- **Mobile Design** — Game elements visibly afford mobile interactions (touching, dragging, scrolling, and/or swiping).

- **Motivated to Continue Play** — After the session, the researcher saw the appeal of the game, regardless of whether the game appealed directly to the researcher. That is, after playing as a naive, interested player using the close reading methodology, this persona would be interested in continuing to engage with the game.
- **NPC Mentor(s)** — As part of the tutorial experience, the player is talked to by an NPC who serves as a mentor figure in some capacity.
- **Narrative Introduction** — The game opens with a story or narrative. This can take the form of a video (cutscene), audio, or even text.
- **On-Demand Info** — The game provides on-demand access to tutorial information, tips, additional educational information, or thorough guides.
- **One Pager** — The game's explicit tutorial is 1–2 pages/screens of basic instructions for controls and mechanics.
- **Performance Benchmarks** — The game provides a way for the player to check their performance.
- **Puzzle Design** — The game has good puzzle design — at least 2 indicators of good puzzle design (see the codebook for indicators).
- **Separation of Tutorial and Game** — There is an explicit tutorial separate from the game itself. For citizen science games, the tutorial is explicitly separated from the scientific activity.
- **Signaling** — The game assists the player's perception while tutorializing through visual cues to draw the player's eye.
- **Standards or Conventions** — The game makes significant use of standard controls or genre conventions in a way that makes the game easier to learn.
- **Strategies Taught** — The game's onboarding / tutorial explicitly introduced or recommended strategies for gameplay, such as tips for when to use a mechanic or guidance on higher-order thinking related to the game.
- **Systems Exploration** — The game actively encourages learning through exploring its systems and interactions.

- **Task Variety** — The game's tasks or experiences vary qualitatively, breaking up the monotony of play.
- **Technical Bugs** — The game has technical bugs, glitches, or other software issues.
- **Tooltips** — The game provides supporting information through the use of tooltips on mouse hover.
- **UI/UX Issues** — The player experiences issues with the UI or UX.
- **Unclear Scoring** — The game's scoring system is unclear in a way that significantly impacts the player experience.
- **Unexplained UI** — Some elements of the user interface are never tutorialized and have no in-game descriptions.

Appendix E

Experiential Prompts

Below are the prompts used for the study in Chapter 6.

1. What are you feeling? Why? **What is the game doing to elicit those emotions?**
2. What are you thinking? Why? **What is the game doing to elicit those thoughts?**
3. What is working / not working about the game-user interaction right now?

   What makes sense?

   What is confusing?

   What is frustrating?

   What is boring?
4. What is the design of the level, environment, and/or quest structure doing to affect your experience and learning? Why did the designers make the game world what it is?

   Why are the objectives what they are?

   Why are the elements of the scene what they are and where they are?

   What seems out of place? Why?
5. What themes are emerging from this experience? **What are the patterns in the design?**
6. How is the game instructing you?

   What are its methods?

   Where is it succeeding or failing, and why?
7. Try to infer what the design process was like and what the designers' intentions are. Compare this to what you are experiencing as a naive player.

For every answer, answer **why/how**, and then answer **why/how** again. At the end, synthesize the surface observations into how they contribute to the deeper themes.

Appendix F

Summary of Games and Experiences

Here I summarize each game's close play from Chapter 6. For each game, I and my co-author Kutub Gandhi describe the game's development camp, genre, core gameplay loop, notable design features, and our overall experiences.

F.1 7 Billion Humans

[518] *Commercial Entertainment* — This puzzle-programming game stood out by its well-designed levels that gradually onboarded the player using multiple types of support. In every level, an NPC explains the goal narratively, then text provides a clear, non-diegetic description of the goal, often with a hint. Additional hints were available on-demand, and the UI provided on-demand details for each mechanic. In combination with optional challenges and all levels being skippable, we never felt stuck in this game. Moreover, the puzzles were interspersed with funny videos and quirky NPC dialogue, making this game more fun than frustrating.

F.2 Air Forte

[47] *Educational — Air Forte* is an early work of Blendo Games (primarily Brendon Chung), now an indie game company. This educational game had a loose narrative introduced by comic panels wrapping a series of educational levels. In each level, a plane follows the player's cursor — the goal is to drag the plane into the correct floating answer bubbles, such as "multiples of five" or "countries in South America." The educational content is entirely orthogonal to both the

gameplay and story, and the game had only a minimal tutorial which taught the controls via a UI graphic with text. Perhaps because I was not the target audience, I felt bored by the game.

F.3 Apetopia

[530] *Citizen Science* — This citizen science game is a first-person endless runner with an unexplained but intriguing aesthetic. Using two pages of "How to Play" information, the game informs the player of the simple mechanics and controls: move left and right to collect coins and avoid obstacles. The tutorial followed standards of a simple endless runner, though the UI elements (health, speed) were unexplained.

Every ten seconds of running, though, is broken up with a clear path and two colored gates. The goal of this section is to go through the gate whose color most closely matches the color of the sky. These sections, one can infer, are the citizen science task which provides data about color perception. The oscillation between the intense running and thoughtful science was, we thought, an excellent balance of gameplay and science. The gameplay provides a mental break and physical challenge while the science provides the reverse. Kutub notes that he would have appreciated an explicit explanation of the scientific benefits of playing the game. As it stood, it was enjoyable, but only attention-grabbing for a short time, without much motivation to play beyond the first 10 minutes.

F.4 Baba Is You

[212] *Commercial Entertainment* — This Sokoban-like puzzle game is an excellent representative of gradual onboarding through increasing complexity and learning through systems exploration. In this 2D grid-based game, the player's goal is to get their controlled avatar (*usually* Baba) to collide with a winning object (*usually* a flag) using simple push mechanics. The twist, however, is introduced quickly: the rules of the game ("Baba is you," "Flag is win," "Wall is stop") are part of the level and can be manipulated like other objects. This game is about challenging one's assumptions, yet the level design is carefully crafted to encourage the player to face those assumptions and refine their mental model. This game is perhaps the best exemplar in our dataset of well-designed puzzles.

F.5 Boson X

[231] *Commercial Entertainment* — *Boson X* is a third-person endless runner with a scientific aesthetic. Overwhelmingly, the feeling of this game was one of frustrating difficulty. The controls were simple, but the gameplay required fast-paced, fine-tuned, twitch-precision inputs. I struggled to complete more than a couple of levels and ultimately rage-quit after the music became repetitive and the game provided no support for overcoming the difficult gameplay.

F.6 Breaking Good

[468] *Educational* — This simplistic match-3 game was overwhelming due to its lack of tutorial and unique mechanical and visual choices. Specifically, the game has four views of a cylindrical match-3 board. Scoring and goals are not explained, and a single screen "help" menu hidden in the settings merely clarifies how to manipulate the primary cylinder. Educational benefits are limited to listings of chemical formulas which serve as goals, but there are no other aspects of learning chemistry.

F.7 Cancer Crusade

[340] *Citizen Science* — *Cancer Crusade* is a mobile citizen science game about simulating cancer treatments. Each level shows a pixellated simulation of a tumor and a timeline of treatments by type and intensity: the goal is to create a treatment timeline which mitigates the tumor's growth. Despite the low production value of the game, the tutorial does several clever things to help learning, including good signaling of new UI elements, smart defaults to the tools, and standard patterns of mobile design hand-holding the player.

However, the game is almost entirely a tutorial, introducing new mechanics every level. Increasingly, the cracks appear: it's unclear how the scoring works, there is little feedback on what to do if your strategy isn't working, and there is little detail available on the scientific aspects of the game. The resulting experience was trial-and-error solving, frustration, and an inability to progress.

F.8 ChromaGun

[387] *Commercial Entertainment* — This puzzle game was reminiscent of *Portal* [535] with a unique color mechanic: the player is given a gun by which to paint things, and a variety of objects that act differently based on their color. For example, floating drones could be painted and would then fly over to walls painted similarly. The tutorial felt successful, with simplistic levels

teaching mechanics one by one. Early levels were quick, and these quick successes created an empowering feeling. Later levels, however, were sometimes tedious: making a mistake would require restarting the entire level. Combined with the fact that the game did not introduce new mechanics or complexities, the core gameplay loop became boring. Minor UX issues caused frustration but didn't ruin the overall experience — sometimes it was unclear what color mode the paint gun was in, and it was unclear when doors were locked or unlocked.

F.9 Colony B

[323] *Citizen Science* — This simplistic and relaxing mobile game appeared polished initially (with a clean UI and straightforward buttons), however the game had some issues that led to frustration. Firstly, the game did not start with a tutorial. The tutorial was a separate menu item on the home screen, which some players could potentially miss. Secondly, the game's scoring system was unclear, and it was further unclear how to improve one's strategy. Thirdly, the game did not elaborate on its scientific aims, leading Kutub to ask why he was playing the game. Finally, the game had little variety or complexity, leading quickly to a boring experience.

F.10 Ding Dong XL

[6] *Commercial Entertainment* — This arcade game, from the makers of *ORBT XL* [5], featured fast-paced and simplistic gameplay. The player controls a ball that bounces around the screen while avoiding obstacles and collecting power-ups. The tutorial is limited to a one-page screen displaying the shape of power-ups and obstacles. Despite the lack of a more thorough tutorial, however, the game was straightforward to understand due to its simple mechanics.

F.11 Don't Escape: 4 Days to Survive

[465] *Commercial Entertainment* — This point-and-click horror game started out in a small, safe area (so players could get accustomed to the mechanics) before branching out into a larger open world. The tutorial was minimal, which wasn't an issue since the Kutub was familiar with the point-and-click genre, however this could pose a challenge to an inexperienced player. The game slowly provided story while increasing the area in the open world that the player could meaningfully interact with. These two elements created a sense of curiosity, while limited resources and the underlying tension caused by the horror elements urged the player to press forward.

F.12 EcoBuilder

[357] *Citizen Science* — This mobile game is the most recent citizen science game on our list to have been developed, released in 2021. Although it had good aesthetic polish, there were several UX issues, such as not being able to register for an account and missing sorely needed quality-of-life features for the research levels. We were overall confused by the game: the scoring was unclear and the lack of feedback made it difficult to strategize. Kutub was able to progress through half of the tutorial with a severe misconception of a core game mechanic which made the rest of the game difficult, and both researchers resorted to trial-and-error solving.

F.13 Eterna

[487] *Citizen Science* — *Eterna* is a citizen science 2D puzzle game played in the web browser in which the player modifies RNA base pairs to achieve a target RNA shape based on a simulation of base pair attractions. While the game initially boasts an excellent aesthetic polish — such as tooltips and good signaling — prolonged play shows the cracks in the user experience: tooltips don't move with the camera and some advanced elements are never taught at all.

For Kutub, the game was overall a positive experience: despite the lengthy text-based tutorials, Kutub felt that the levels had clear goals and the hint system helped them multiple times. I, on the other hand, felt overwhelmed by the constant introduction of new mechanics and wanted more opportunities to practice and develop strategies. Indeed, for both researchers, when the initial strategies failed to produce results, they resorted to trial-and-error solving. Our experiences align with Keep [255], who notes that the sheer number of mechanics introduced to *Eterna* over the course of its history results in a disconnect between the initial simple tutorial levels and the frustratingly difficult and unexplained scientific puzzles.

F.14 Eyewire

[408] *Citizen Science* — *Eyewire* turned out to be less of a game and more of a gamified program (though still gameful enough to include in our dataset). The simple task — color in 2D images of neuron microscopy data slices to model a 3D reconstruction of the neuron — is fairly straightforward with a supportive tutorial. The tutorial includes sufficient practice, adequate (albeit non-specific) feedback, and a gradual increase in difficulty. However, only minutes after the tutorial, the game's novelty wears off, and the only motivating gamification is a leaderboard and several achievements for continued engagement. Kutub felt frustrated by the game's AI which automat-

ically translates 2D coloring to 3D reconstruction: he was often unsure what the AI would do in response to his input. In addition to a plethora of untaught advanced tools, this game felt more like gamified software and was, although relaxing, quickly boring.

F.15 Foldit

Note: I am a developer on Foldit, so this section represents Kutub's views.

[531] *Citizen Science* — This game is one of the earlier examples of a citizen science game, and is filled with features from its long history of development. The actual citizen science aspect involves protein folding, with the intellectual work being done by the player, and tedious work automated through computer assistance.

Foldit had an immense amount of complexity, and was overall an overwhelming experience, however there were a variety of tutorial aspects that aided the onboarding process. Gamification elements (getting points for solving a level, sounds, flashing lights, and simulated fireworks) and interest in the science (cultivated by discussions on the scientific purpose of the play) motivated the researcher to continue playing.

The tutorial itself was primarily text-focused (i.e., it relied a lot on text boxes explaining what to do), however it was well-written. The length of the tutorial meant that the researcher did not finish playing through it, however they did attempt to play the non-tutorial puzzles anyways. The non-tutorial puzzles were significantly more difficult, and the researcher found it difficult to progress, leading to disappointment.

F.16 FTL: Faster Than Light

[493] *Commercial Entertainment* — This roguelike strategy game simulates the experience of being a spaceship officer. Rather than aiming and firing weapons yourself, you give orders, balancing the resources you provide to systems for the offense, defense, repair, etc., of your ship. The tutorial and story were minimal, however this shortcoming was balanced by incredibly well-written tooltips explaining every aspect of each system. The initial combats and tooltips allowed Kutub to understand basic strategy and mechanics quickly.

F.17 Game Dev Tycoon

[193] *Commercial Entertainment* — This game simulates running a game development company, making decisions about your game's genre and topic and how your team allocates de-

velopment time. Although I was confused about how to strategically make choices, this was mitigated by the game's concept being grounded in a real-life, known subject that I could reason about. Like several other commercial games, *Game Dev Tycoon* featured a gradual increase in complexity. Overall, the experience was learnable and enjoyable.

F.18 Gunpoint

[498] *Commercial Entertainment* — In *Gunpoint*, a commercial 2D stealth/puzzle game, you play as a spy infiltrating various buildings to steal objects or information. The tutorial implemented several helpful learning patterns: on-screen contextual controls, tooltips on mouse hover, a safe introduction of mechanics before challenging the player, clear design language, and a gradual increase in complexity. The level design in particular scaffolds learning by introducing mechanics slowly and then integrating them into more complex challenges. Each level was also introduced narratively which provided variety and explained the goals and mechanics. *Gunpoint* was easy to learn and fun to play.

F.19 Ikaruga

[522] *Commercial Entertainment* — This shoot-em-up game was originally developed in 2001 for Japanese arcade machines, and the PC adaptation clearly reflected this. After a bit of initial confusion, I navigated to a How to Play mode which introduced on-screen controls as text, including the original arcade instructions, and it illustrated the concepts with an animated simulation of gameplay. Although the game did not tutorialize any strategies, the primary difficulty is one of dexterity, not strategy, so I did not feel lost for what to do.

F.20 Influent

[433] *Educational* — This educational game advertises itself as a language learning game, yet this led to disappointment in the mismatch of expectations. At first, *Influent* appeared very polished with an interesting narrative animated video. However, it transitioned into a frontloaded tutorial that taught all controls and mechanics at once, rather than just-in-time. The instructions were overwhelming and unclear, and the UI was confusing since elements were either unexplained or the explanation was lost in the deluge of information.

The actual gameplay was about matching pronunciations and written words to virtual objects — rather than being scaffolded into learning a language and its grammar, the game was merely practice for listening and identifying words. I chose to practice Korean — a language I don't know — and did not retain any knowledge after the rote memorization involved in performing the game's tasks.

F.21 Kerbal Space Program

[485] *Educational — Kerbal Space Program* is a commercial simulation game that lets the player craft and fly space-faring vessels. Its tutorial, though optional and separated from the main flow of the game, is well-designed to hand-hold the player through interactions and mechanics step-by-step. The tutorial is separated into individual modules that focus on one aspect of gameplay at a time, guiding the player through complicated user interfaces and workflows. Moreover, the tutorial modules are a controlled, designed space: player interactions are limited, only relevant elements are introduced, and just-in-time reminders are provided for controls as needed. The tutorial is text-heavy, but this is ameliorated by an NPC mentor with a flavorful personality. Other features that scaffold the learning include tooltips on hover and authentic scenarios to ground the training. Overall, despite *Kerbal Space Program* being complex, Kutub felt reassured by the game: the initial interactions instilled a sense of trust that there would be a helpful tutorial for all of the game's mechanics. This feeling was reinforced by the tooltips and in-game manual.

F.22 Lazy Galaxy

[90] *Commercial Entertainment — Lazy Galaxy* is primary an idle game, though somewhat more involved than exemplars of the genre (e.g., *Cookie Clicker* [245]). The game asks the player to make (relatively) complex decisions regarding which resources and mining approaches to prioritize. The resource-gathering section was punctuated with automatic combat sections, where your AI-controlled ships invaded other alien species.

The tutorial of this game often told you what to do, but not how or why. This confusion on "how" was exacerbated by a UI composing mostly of alien symbols rather than English text. There was a lack of feedback regarding the correctness of one's approach — once automated machines were placed, resources were collected, but it was unclear whether the player strategy was optimal, or what benefit optimality would even bring. The combat sections were a nice change of pace, however they were strategically trivial. You were given the option of which ships to send out, but there was seemingly only one reasonable choice to make — after which the player would then simply sit and watch the combat play out. While it is plausible that future combats would be more complex, these were not experienced in the onboarding of the game.

F.23 Learn Japanese to Survive! Kanji Combat

[480] *Educational* — Similar to *Influent* [433], *Learn Japanese to Survive! Kanji Combat* is advertised as an educational, language learning game. However, unlike *Influent*, its purpose is made more specific and explicit: the game only attempts to teach Japanese kanji. This is made clear both in the game's marketing and at the start of the game itself. Then the game has about ten minutes of a visual novel introduction to the story, building anticipation to the learning material and diegetically explaining why the game is about learning kanji.

The gameplay itself is a balance between kanji lessons introducing new material and a JRPG for practicing the material. The lessons introduce new kanji using several scaffolding techniques, including signaling, limited text per screen, visualizations, opportunities for practice, and a check for understanding at the end. The JRPG side of play tests whether the player can match kanji to definitions and vice versa by having battles with kanji that can be defeated by using the appropriate definition.

Learn Japanese to Survive! Kanji Combat is a well-polished game that uses standards of the JRPG genre for reinforcing learning. However, the material is introduced at a fast pace and more information is taught than is tested by the RPG mechanics. Moreover, the gameplay is increasingly repetitive in its sound effects, and navigating the UI to find the answer you're looking for becomes difficult as more kanji are introduced.

F.24 Lightmatter

[524] *Commercial Entertainment* — This first-person puzzle-platformer has similar gameplay to *Portal* [535] and *Chroma Gun* [387]. However, rather than a gun which manipulates the environment, the player can hold or place lamps which light up an area. This is important because the core mechanic is that the player cannot walk across dark areas.

The tutorial itself is very minimal: instead of controls being taught, the game assumes the player has the game literacy to make sense of the game based on existing genre conventions. Onboarding happens through environment design and visual affordances of the puzzle elements, such as buttons and climbable ledges. An NPC provides diegetic explanations for the puzzles and dialogue between puzzles to vary the player's attentional load. The puzzles themselves have clear goals, clear affordances, and clear first steps of interaction. Overall, the onboarding of the game works very well for players who are familiar with the genre conventions but may be less helpful for novice players.

F.25 Luxor Evolved

[348] *Commercial Entertainment — Luxor Evolved* is a fast-paced arcade game. The core gameplay is about shooting orbs of different colors to match and destroy incoming orbs in a unique blend of tower defense and match-3 mechanics; although difficult to describe, the gameplay is simple to understand when one starts playing. The tutorial was minimal, but this was not an issue as the game was easy to grasp. The one exception to this was the power-ups system: power-ups would sometimes fall towards the player from the top of the screen. Kutub initially thought these were to be avoided, and even when realizing they were power-ups, was confused about the specific benefits of the various power-ups. Information on the power-ups was available via a “how to play” menu hidden within the settings, however the researcher did not find this on their initial playthrough.

F.26 Lyne

[510] *Commercial Entertainment — Lyne* is a minimalist puzzle game about connecting nodes of different shapes such that lines start at the start node, go through all nodes of the same shape, end at the end node, and do not cross other lines. The onboarding is similarly minimal with only a few words of text and a few animated gestures to demonstrate how to interact with the game. The rest of the onboarding comes from a gradual (indeed, a very slow) increase in complexity and a clear visual design language.

F.27 Mini Metro

[138] *Commercial Entertainment — Mini Metro* is a minimalist mobile game about drawing metro lines to connect stations efficiently. The game uses animated gestures and signaling with dynamic help text to onboard players into the basic interactions, then gradually increases in complexity. *Mini Metro* follows the common design patterns of mobile games in terms of tutorializing controls. The calm music, minimalist aesthetic, and gradual difficulty ramp makes *Mini Metro* a pleasant game to learn and play.

F.28 Mozak

[532] *Citizen Science — Mozak* is a citizen science game about 3D neuron reconstruction

played in the web browser. In this way, it is very similar to *Eyewire* [408], except rather than using 2D image slices, Mozak presents the data directly on top of the 3D visualization and asks the player to reconstruct via tracing. *Mozak* is more of a gamified task than a full-fledged game (though still gameful enough to include in this study), and most tools and controls did not have a tutorial. We experienced bugs and frustrations with the user experience and UI, such as issues with the camera constantly resetting, and the task itself was monotonous.

F.29 Niche, a genetics survival game

[532] *Educational* — *Niche* is an educational game about genetics. It starts with a narrative animated video introduction before jumping into a text-heavy tutorial about how to control your animals as they search a hex-based island for food and mates. When your animals have a child, you can select their genes from a large array of unexplained options. Our experiences with *Niche* were one of confusion: is this a simulation game or a strategy game? Should I be thinking strategically about how to select genes, or is this a game to learn about genetics through playful experimentation? The game gave little information for making intelligent, strategic decisions, and yet the difficulty was intense: your animals starve within a few turns if you can't find food. In combination with a frustrating user interface that took several clicks for each interaction, we felt bored and confused by this maybe-strategic, maybe-educational roguelike game.

F.30 Odyssey — The Story of Science

[508] *Educational* — This first-person educational game is about exploration and puzzle-solving. The game opens with a narrative introduction: you arrive on an island while someone tells you over the radio that they need your help. You then find a journal containing a child's notes on meteorological and physical phenomena, documenting what they learned from their scientist father about, as examples, how to find the north star and why people on the other side of the world don't fall off.

This game features strong intrinsic integration between the educational content and the puzzles. The player must read the journal to understand the natural phenomena and solve environmental puzzles related to them. Between exploring and puzzle solving, the game has a good balance of tasks and the environment is well-designed to structure exploration. Although we experienced minor issues with the camera controls and were easily bored by the educational material, we believe this game would be very engaging for the appropriate learner level (approximately 5-8^{th} grade).

F.31 Orbt XL

[5] *Commercial Entertainment — Orbt XL* is a simplistic arcade game where one controls a planet orbiting a black hole and must avoid falling asteroids and planets. By orbiting faster or slower, the player can move closer or further from the black hole; this was the only control the player had. The game told the player they could click to orbit faster and to avoid obstacles — this was the extent of the tutorial, however, this was all that was necessary. Obstacles that appeared later occasionally had different properties, but their mechanics were easy to infer through observation. One could collect power-ups, and this was never explained, however the power-ups were distinct enough from the obstacles such that it was obvious they were meant to be collected.

F.32 Opus Magnum

[577] *Commercial Entertainment — Opus Magnum* is one of Zachtronics' puzzle-programming games, published after *SpaceChem* [574], *TIS-100* [575], and *Shenzhen I/O* [576]. In this entertainment game, you play as an alchemist arranging machine components on a hex grid which can be programmed in a series of steps to produce molecules from atoms. *Opus Magnum* scaffolds the player in several ways: narrative dialog, tooltips, signaling, and leaderboards that let you compare your performance to other players on each puzzle. Although the tutorial levels introduce mechanic after mechanic without practice, the game overall has a gradual increase in complexity — once the player understands the basic mechanics, the puzzles are increasingly more complicated. Notably for onboarding, the first level has no goal except to see a working solution, while the second level provides working examples and simply asks the player to make a similar copy, thus walking them through initial performance. Despite its complexity, *Opus Magnum* was fairly accessible, especially because one can brute-force their way through any puzzle (using the unlimited space and time provided) if they can't figure out a more efficient solution.

F.33 Papers, Please

[1] *Commercial Entertainment — Papers, Please* is a game about being a border-crossing immigration officer for a fictional country. During play, the player checks travelers' documents to determine whether they should be allowed to enter based on a changing list of rules. Like many other commercial games, *Papers, Please* has a gradual increase in complexity. Despite most travelers being random, the game seems to have scripted onboarding for when a new rule is introduced: in my playthrough, the first traveler always followed the new rule and the second traveler always broke

it. This allowed for immediate practice. Moreover, failing a rule led to immediate feedback via a warning, or reduced pay on subsequent failures. The player's compensation is important because there is only a limited amount of time each day to earn money for travelers processed, and money is spent on taking care of the player character's family. The resulting experience felt like what the developer, Lucas Pope, seems to have intended: I felt the pressures of a border worker, frantically trying to assess whether each traveler was a threat to the country and where I could cut corners to save money for my family — while perhaps not an enjoyable feeling, *Papers, Please* is a eudaimonic masterpiece.

F.34 PC Building Simulator

[88] *Educational* — In this first-person job simulator, the player takes on the role of a computer repair technician starting their own business where they take apart computers and replace parts for clients. The game assumes some gameplay literacy from the player but provides on-screen controls, an explicit tutorial that hand-holds step-by-step interactions, and highlights 3D objects to show their affordances (for example, when mousing over a screw, the screw is outlined to show that it can be removed). The onboarding is supported by the fact that the game's simulation is grounded in a real life task, so players with familiarity in PC building can apply their knowledge and expectations to the game.

F.35 Phylo

[322] *Citizen Science* — *Phylo* is a citizen science puzzle game about aligning similar sequences of DNA to try to figure out which base pairs were added, removed, or changed from the original. There were three modes of interest to us: the normal game, the tutorial, and a Story mode. The tutorial uses the common method of text boxes to explain various UI elements, although it remained unclear how the scoring system worked and how to use the game mechanics strategically. The information was sufficient to get us into the normal game and able to complete a couple of levels, but shortly thereafter the game ramped up quickly in difficulty — without a sense of strategies to employ, the puzzles are easily a mess of trial-and-error.

The Story mode came with its own tutorial, and although the text explained elements better than the original, this tutorial was non-interactive (a set explanation rather than a series of tutorial puzzles with guiding prompts). Not only did the Story mode duplicate the tutorial, but loading screen tips between levels also provided redundant information. Story mode was additionally interwoven with comic panels of a science fiction narrative and real-life quizzes that tested the player on material not taught by the game. Overall, this game was difficult and confusing.

F.36 Plague Inc: Evolved

[356] *Commercial Entertainment* — In *Plague Inc: Evolved*, you play as a plague attempting to infect the world through strategy-simulation tactics of evolving to be more contagious and lethal while evading the discovery of a cure. The game features an explicit tutorial which hand-holds the player through a whole scenario and explains each UI element while signaling to it. Although there is additional on-demand information available, most of the game is learned through systems exploration. Over the course of play, I discovered a decent strategy and felt that I was successful for the first half of his first game. However, in the late game, one country shut down and my plague was unable to reach it. This led to a boring, slow failure where it was unclear at what point the game became unwinnable, and further unclear what I could have done better to win.

F.37 Poly Bridge

Commercial Entertainment — *Poly Bridge* is a straightforward bridge-building simulation/puzzle game. The physics and UI were intuitive (and there were tooltips on the various buttons), though the tutorial was short and Kutub was not confident in his skills as they exited the tutorial. As Kutub continued his play, he grew frustrated that the tutorial had not shown the full power of the tools that the game provides; i.e., the tutorial showed the usage of the tools in a simplified level, but the player did not have a deeper understanding of how they should be applied as preparation for the later levels. Furthermore, the later levels felt repetitive and tedious. This led to boredom, and a fear that the researcher's skills were not improving in preparation for levels even further on.

Despite these specific criticisms, the Kutub enjoyed playing this game and found it overall well-designed.

F.38 President for a Day — Floodings

[469] *Educational* — *President for a Day — Floodings* is tagged as Educational on Steam, though in practice the only educational aspect is perhaps providing some background on the geography of Pakistan. In this game, you play as the president of Pakistan making decisions about how to allocate limited resources to deal with floods, famine, cholera, rebels, and refugees. Like many games, the tutorial is a series of text boxes that point to UI elements and explain how they relate to

your overall goal. Yet, in this game, there is a timed pressure to act and the tutorial doesn't pause the game. This led to me skimming the frontloaded information under pressure while trying to understand enough to play with the time I had left. At the end of the game, I lost and it was unclear what I could have done better.

F.39 Project Hospital

[373] *Educational* — *Project Hospital* is a management simulator game where you build a hospital. The amount of options and control you have over fine details was incredible, though not overwhelming initially since it was clear most of it could be ignored. Unfortunately, the tutorial fell apart rather quickly, with a lack of feedback on whether the right decision was made, and with a rapid bloom in complexity. Furthermore, the tutorial did not explain why certain decisions were made and skipped steps in the explanation. The second phase of the tutorial led to even more frustration. One cannot go back to a previous step in the tutorial without restarting the whole level, certain UI elements were unexplained, and at certain points a trigger needed to be reached before one could progress — but sometimes it was unclear what was blocking the trigger from completing.

F.40 Quantum Minds

[462] *Citizen Science* — This game had a lack of sound and a simplistic UI that indicated it was built with a low budget. The mechanics were intuitive since the primary gameplay loop (guide a yellow liquid to a goal area) featured a realistic physics system. Each level required some amount of pathing strategy and dexterous execution. Completing a level required succeeding three times in row, a mechanic that forced Kutub to be consistent and demonstrate mastery of each level. At the same time, it also limited his desire to experiment with different paths. On later levels, this mechanic led to frustration and stress — if Kutub had worked hard to complete the level twice in a row, failing the third time was a major setback. Finally, it is worth noting the game did not explain its scientific goals, which frustrated Kutub.

F.41 Quantum Moves 2

[464] *Citizen Science* — *Quantum Moves 2* is a citizen science game about quantum physics, built by the same developers as *Quantum Minds* [462]. In this game, the player drags a reticle which manipulates some sloshing liquid; the goal is to get the liquid to a specified place in limited time. It was unclear how this gameplay relates to quantum physics and I wanted to know

more about the connection to science. The tutorial levels signaled to specific elements, blocking out elements that the player didn't need yet, and explained briefly what to do. However, by the end of the tutorial, the gameplay and strategy were still unclear. In combination with a lack of feedback and no way to get more information about the UI, this led to trial-and-error solving and a frustrating, confusing experience.

F.42 Skill Lab: Science Detective

[463] *Citizen Science — Skill Lab: Science Detective* is a suite of mini-games used to measure cognitive abilities at population scale for citizen psych-science research. The game is clearly research from the beginning: there is a significant sign-up and consent process to start playing. Once inside, an NPC mentor walks the player through navigating to each mini-game. Each mini-game has its own tutorial that directly explains how to play, albeit with a lot of text. The mini-games are gamified tasks, such as the game of Memory or spatially aligning a 3D object to match a shadow of it at a given angle. Each mini-game has a gradual increase in complexity and is fairly easy to pick up and play. Kutub wanted to know more about the science behind the games and there were a couple of minor UI/UX issues experienced, but *Skill Lab: Science Detective* was otherwise a playable and (briefly) entertaining set of arcade games for a useful scientific purpose.

F.43 Sokobond

[8] *Educational — Sokobond* is a Sokoban-style commercial entertainment game with a minimalist scientific aesthetic. In this puzzle game, you control an atom and, by moving in the grid-based level, try to form a given molecule by connecting to other atoms. The game features a good aesthetic polish, a clear visual design language, and a gradual increase in complexity. We even noticed that the control prompts adapt to whether the player is using a gamepad controller versus keyboard. *Sokobond* is tagged as Educational on Steam because after each level the player is given an educational trivia blurb about the molecule they created, such as "Methane: CH_4 Earthquakes can release methane into the ocean from undersea reserves." These trivia facts were not integrated into gameplay and easily forgotten; in fact, we found it very easy when playing to accidentally click through the fact at the end and would need to replay the level if we wanted to see the trivia again. Moreover, by grounding the gameplay in science, Kutub felt unsure if he was making the molecule correctly, for example by connecting the atoms at the correct angles. Overall, this game was very approachable (though some levels excruciatingly frustrating in difficulty) with interesting

but forgettable educational trivia.

F.44 Starbound

[80] *Commercial Entertainment* — This 2D action-RPG reminded Kutub of *Terraria* [424]. Interesting animations and visuals spurred the researcher to continue forward. The controls were straightforward and the tutorial provided just-in-time reminders of certain elements. There was some confusion about the inventory system, for example the clothing UI was complex and it was unclear how to equip certain items into the hotbar (in addition to assuming gameplay literacy that one knew what a "hotbar" was). As the game opened up, there were some further frustrations with the lack of reminders and UI indication (e.g. reminders on how to use one's flashlight and the medikit), and frustrations with the lack of direct objectives, however the experience was overall pleasant and inviting.

F.45 Super Hexagon

[507] *Commercial Entertainment* — This arcade game was a very brief experience since the entire game is about quick reflexes and a simple premise. In this colorful, flashing game, hexagons with sides missing close in on the center of the screen where a triangle points to one of the hexagonal angles. The goal is to rotate the triangle so that it never collides with a wall of the incoming hexagons. The game assumes some gameplay literacy and provides only minimal on-screen controls. Ultimately, I found the game to be very difficult, surviving mere seconds each time, and thus felt bored because I was not improving over time.

F.46 The Room Three

[162] *Commercial Entertainment — The Room Three* is a point-and-click escape room puzzle game. The tutorial is seamlessly integrated into the game, teaching camera controls and basic interactions using on-screen prompts and animated gestures and gradually releasing the hand-holding instruction. However, beyond the introduction, the affordances of the environment were unclear: I was unsure what could be clicked on or interacted with. This led to the puzzles not being difficult for their logic, but because the user interface masked how to even engage with the game. Moreover, although there was a hint system, it felt like cheating — especially because it wasn't the puzzles that were difficult to solve. This game was frustrating and confusing, though these feelings could have been easily mitigated with a better visualization of affordances.

F.47 TIS-100

[575] *Educational* — *TIS-100* is a commercial puzzle-programming game by Zachtronics. In this game, the player solves a series of programming puzzles in a fictional assembly language. The game is clearly designed for Zachtronics' hardcore puzzle audience, given that the tutorial is provided as an accompanying PDF and is made to seem like a digitized print-out of a manual book. Thus the game assumes gameplay literacy, requiring that the player read an esoteric manual and navigate the game's UI on their own to find and solve the puzzles.

The gameplay itself features bits of narrative throughout each level (in the form of mysterious error messages), a gradual increase in complexity, and performance benchmarks for the player to check their understanding (in the form of global leaderboards). Similar to *Opus Magnum* [577], the first level shows the player a successful example and simply asks them to make a similar solution, demonstrating what correct use of the mechanics might look like. Overall, since I am familiar with assembly programming, this game was accessible and fun for me, but would likely be inaccessible to players without prior programming knowledge.

F.48 TowerFall Ascension

[155] *Commercial Entertainment* — This commercial game is primarily designed as a couch co-op action game, but has a standalone single-player mode. It was reasonably simplistic in its controls, though certain enemy patterns and abilities felt somewhat complex. The tutorial was similarly simplistic and didn't explain some important game elements (e.g., the play area wraps around itself like in *Pac-Man* [354] and enemies can be jumped on to deal damage like in *Super Mario Bros.* [363]) though Kutub notes that the couch co-op nature of this game affords a lack of instruction — players can enjoy working with friends to learn these elements of the game, and players who are more experienced can explain the intricacies to their less-experienced friends.

F.49 Tyto Ecology

[232] *Educational* — *Tyto Ecology* aims to teach about ecosystems through a simulation of species interactions. The player takes on a god-like role in placing species in an ecosystem and can watch the results over time. The tutorial hand-holds the player through camera controls, basic interactions, and core gameplay mechanics, though some of the UI elements are never explained. A "biodex" provides further on-demand educational information for each species in the game. Overall, we were unsure whether this game wanted to be a pure simulation or a strategy game: each species

had various mechanical statistics that were not explained how they interact with each other. Moreover, the pacing of the game (waiting for ecosystem growth to happen) feels very slow, although you can choose to instead time-skip by large gaps — and risk having your system die out from lack of oversight. The gameplay was thus boring and confusing, although we found small charms in the cute animals (before they were devoured by predators).

F.50 Unheard — Voices of Crime

[359] *Commercial Entertainment* — This commercial entertainment game places the player in a detective role with the goal of solving a crime. In each level, the player sees a video recording of a top-down abstraction of the crime scene and can move an avatar to eavesdrop on specific areas of the level at specific moments in time. In this way, gameplay is about piecing together clues to try to put names to unknown individuals and track who did what when. The tutorial uses good signaling in combination with animated gestures and text instructions to walk the player through basic interactions. Then the game provides a small practice level for the player to get acclimated. Each level has a clear goal (answer certain questions about the crime) and a clear first step (listen into all conversations). However, the game can get repetitive since there is no way to increase the speed of the audio and the player may need to listen to conversations multiple times. Moreover, since listening is the majority of gameplay, the task can feel monotonous and non-interactive.

F.51 Veritas

[184] *Commercial Entertainment* — This point-and-click horror puzzle game featured plenty of interesting puzzles, many of which were presented at the same time (with the benefit being that, if a player was stuck on one puzzle, they could always pivot to another). Kutub did not enjoy the game, however, they conceded that he would have enjoyed it if he was looking for this kind of experience (horror/puzzle). The slow interactions led to a sense of tedium, and the fact that you couldn't directly control your character (only click doors or arrows to go to an entirely different view) meant that it was hard to get a sense of space and direction.

F.52 War Solution — Casual Math Game

[571] *Educational* — In this match-based math action game, your goal is to defeat your (AI) opponent by solving math problems faster than they can. Each player has a castle and answering the multiple choice question prompt would allow you to catapult boulders at enemy fortifica-

tions. There were a variety of minor UI issues and UX annoyances initially. The gameplay itself features a variety of gamified mechanics (such as different power-ups) that were explained via an overwhelming amount of text. In regards to the educational aspect, the math problems were unrelated to the gameplay, leading Kutub to question how the game taught math better than a simple worksheet. In fact, Kutub found that a winning strategy was clicking randomly on the multiple choice questions, completely ignoring the math involved. Furthermore, there was no ability to change the difficulty or types of questions asked, so it would only be beneficial to students drilling a specific level of arithmetic questions. The gameplay outside of the educational aspects was shallow: there were power-ups and combat mechanics, but the decisions at every step were trivial.

F.53 while True: learn()

[307] *Educational* — This educational game tries to teach core concepts of machine learning through a puzzle-programming game with a cat-focused aesthetic and narrative. Notably, the game featured screens that explicitly draw the connection between in-game mechanics and real world implications while including links to resources for further reading. The levels were structured in a branching path such that the main path continuously introduced new mechanics through trivial gameplay while the branches offered a variety of tasks and gamification systems. The puzzles were also separated by bits of narrative and tangential education on the history of machine learning. As a game, *while True: learn()* is often either trivial and introducing new mechanics or quickly ramping up in difficulty. As education, it provides a high-level overview and some historical trivia without diving into the details of machine learning programming.

F.54 Zoombinis

[506] *Educational* — This remake of the classic educational game from 1996 comes from an earlier wave of edutainment software. The game opens into a lengthy narrative cutscene that introduces the story, followed by a character creation screen where the player can customize their team of 16 Zoombinis. After that, the player embarks on a series of puzzles, each one more similar than the last. In every puzzle, there is a hidden, arbitrary rule that can only be solved by trial-and-error, usually in the form of sorting your Zoombinis. For example, the first level features two bridges: Zoombinis can only cross one bridge or the other based on a feature of their bodies. However, this rule can only be discerned through trial-and-error until there are enough examples that one can apply inductive reasoning. *Zoombinis* has little in the way of an actual tutorial, instead

assuming gameplay literacy (in 1996!). We found the lack of clear feedback and support to be frustrating. While this game may teach a child inductive reasoning skills, it would be primarily from the child's own persistence that they learn.

F.55 Zup! 2

[412] *Commercial Entertainment — Zup! 2* is a short, minimalist puzzle game about the careful interactions of a few simple pieces. The game features only a handful of elements: red blocks which explode when clicked, yellow blocks which physically respond to explosions, a ball which also responds to physics, a green platform which wins the level if the ball rests on it, and purple blocks which are immune to physical manipulation. The remainder of the game happens through level design and learning through systems exploration: there is no tutorial or explicit instructions of any kind; in fact, aside from the word "Zup," the entire game is non-verbal and never addresses the player. The player learns, through a series of levels, how to explode the red blocks with proper timing to create chain reactions that move the ball onto the goal platform.

Appendix G

Design Documentation

This section describes some of the theoretical grounding involved in the design process when designing new *Foldit* tutorial levels. This work is based on 4C/ID (e.g., [540]) and GEL (e.g., [87]), among other pedagogical concepts and frameworks (e.g., [42]).

G.1 Learning Tasks

These are the learning tasks of *Foldit*, as determined through a combination of empirical qualitative research and my own understanding of the game, generated via CTA.

- Design
  - Early game
    - Planning
    - SS assignment
    - Rough hand-folding
  - Mid game
    - Shape refinement
    - Amino acid and bond selection
    - Issue cleanup
  - Late game / evolving
    - Recipe running

    * Fine-grain refinement

- Prediction
  - Early game
    * Identification and planning
    * Rough hand-folding
    * SS re-assignment
  - Mid game
    * Shape refinement
    * Issue cleanup
  - Late game / evolving
    * Recipe running
    * Fine-grain refinement
- Electron Density
  - Early game
    * Finding landmarks
    * Cutting and arranging
    * Using smaller landmarks
  - Mid game
    * Revising
    * Issue cleanup
  - Late game / evolving
    * Recipe running
    * Fine-grain refinement
- Small molecule design
  - Early game
    * Planning
    * SS assignment

    * Rough hand-folding
  - Mid game
    * Shape refinement
    * Amino acid and bond selection
    * Issue cleanup
  - Late game / evolving
    * Recipe running
    * Fine-grain refinement
- Binder design
  - Early game
    * Planning
    * SS assignment
    * Rough hand-folding
  - Mid game
    * Shape refinement
    * Amino acid and bond selection
    * Issue cleanup
  - Late game / evolving
    * Recipe running
    * Fine-grain refinement
- Symmetry
  - Early game
    * Planning
    * SS assignment
    * Rough hand-folding
  - Mid game
    * Shape refinement

        * Amino acid and bond selection
        * Issue cleanup
    - Late game / evolving
        * Recipe running
        * Fine-grain refinement
- Group work
    - Joining a group
    - Sharing a solution
    - Evolving a solution
- Research
    - Articulating a question
    - Choosing a database
    - Searching with keywords
    - Finding information
    - Integrating with knowledge
- Recipe-writing
    - Planning and outlining
    - Looking up syntax
    - Testing and debugging
    - Publishing

G.2 Design Principles

These are the design principles used in development, drawing from the theories as stated at the beginning of this Appendix, but primarily from 4C/ID.

- **Sequencing**: Order levels from simple to complex.
- **Whole task**: Levels should be oriented toward skills that help with "whole tasks," i.e., puzzle types.

- **Training wheels**: When learners are not yet ready for a whole task, supports and constraints should be used to help them perform whole tasks before competence.
- **Completion-strategy / fading guidance**: When introducing new procedures, have the learner first practice with an annotated partial solution before fading guidance over one or more practice levels.
- **Prior knowledge activation**: Before teaching something new, recall relevant prior knowledge.
- **Dynamic visualization**: Use animated demonstrations to show step-by-step procedures or concepts with a temporal component (e.g., pulling).
- **Redundancy**: Don't provide information redundantly, or allow recaps to be skippable.
- **Coherence**: Don't include non-essential details, or allow extra information to be skippable.
- **Self-explanation / Butterfly**: Through task design, encourage users to slow down and process information deeply; don't let users come to the wrong conclusions by fluttering around or going through too fast.
- **Temporal and spatial split-attention**: Keep related items close to each other temporally and spatially.
- **Signaling**: Direct the player's attention when you want them to be focusing on something.
- **Segmentation**: Segment learning into meaningful units.
- **Component-fluency / Strengthening for common routines**: Include additional practice for specific sub-procedures if and only if learners need to be strongly skilled in those sub-procedures.
- **Individualization**: When sensible, adapt the instruction to needs of learner.
- **Second-order scaffolding**: Over the course of learning, gradually transition from telling the learner what to learn to letting them direct their own learning.
- **Motivation**: Motivate learning by emphasizing benefits of learning the new material as it relates to authentic problem solving and personal benefit.
- **Novelty and challenge**: Learning should gradually increase in novelty and challenge.
- **Worked examples**: Include worked examples for learners to learn from.
- **Varying examples**: Examples provided should be as diverse as possible while highlighting key differences and similarities to attend to.
- **Backwards chaining**: When teaching a complex procedure, start by teaching the last step, then teach the second-to-last (and last), and work backwards toward the first step of the procedure.

- **Variable practice / Performance transfer**: As much as possible, vary the nature of the problems presented for better learning transfer.
- **Rule-formation testing**: Anticipate common misconceptions, mistakes, and flaws of generalization/specification; challenge the learners on each of these flawed mental models one at a time to ensure that the mental model they form is correct and nuanced.
- **Targeted feedback**: Provide clear, relevant, immediate, thorough, individualized feedback to the learner's successes and mistakes.
- **Clear goal**: Each level should have a clear goal.
- **Clear first step**: Each level should have a clear first step to take.
- **Multimodal presentation**: Information should be provided verbally through audio and non-verbally through video.

G.3 Template for Lesson Design

This is a lesson template used for guiding level design, drawing from the theories as stated at the beginning of this Appendix, but primarily from GEL.

G.3.1 Lesson Objective

Action/Trigger/Standard

What actions will learners understand after this lesson? What triggers cue that these actions should be performed? What are the standards of measuring success of these actions?

G.3.2 Reason for Lesson

Benefits / Risks of not learning

What value does this lesson provide? What does the learner risk by not having this lesson?

G.3.3 Overview of Lesson

Relate to prior knowledge / prerequisite skills

How does this lesson relate to their previous learning?

Identify unusual elements

What key and unusual elements should learners attend to?

Position into overall training

How does this lesson fit into the whole task?

Instructional strategies

How will learners be taught in this lesson? What will learners be expected to do in this lesson?

G.3.4 Necessary concepts and processes

Connect by analogy

For each new concept, explain by analogy to connect to prior knowledge.

Conceptual knowledge

Provide necessary conceptual knowledge.

When and how

For each procedure, when and how should the procedure be performed?

Step by step procedures

For each procedure, how do you perform it step-by-step?

G.3.5 Demonstration

Provide a clear step-by-step demonstration of the actions to learn.

G.3.6 Practice with feedback

Learners practice with as detailed and thorough feedback as possible.

G.3.7 Example

- This Tutorial will teach you how to form disulfide bridges.
- Do you want to know more about the science of the term “disulfide bridge”?
  - ***On ‘yes’:*** Disulfide bridges, or disulfide bonds, are the bond between two sulfur atoms: di-, meaning two, and sulfide, meaning sulfurs. There are two amino acids that have sulfur atoms, cysteine and methionine, but only cysteine’s sulfur atom is reactive enough to form disulfide bonds.
- Disulfide bridges are even stronger than hydrogen bonds, so knowing how to create them in your proteins can help your proteins keep their shape and synthesize in the lab!

- Creating disulfide bridges happens during the early or mid-game, while drafting your fold. At the same time that you're making hydrogen bonds, you should look for places to form disulfide bridges.
- Disulfide bridges can be formed when two cysteines can be bonded together.
- You'll know you've bonded two cysteines when you see a green and white bond.
- Look at this disulfide bond. Notice that the cysteines are at a right angle to each other. This angle is important for forming a good disulfide bond.
- This is also a good distance for the bond to form. Too close or far away and the sulfur atoms won't connect. The best way to form a disulfide bridge is to band cysteines together.
  - Try it yourself! Right click and drag between the sulfur atoms of these two cysteines to make a band.
  - ***On bad band:*** *[Log this mistake for gameplay performance analysis.]* Close, but not quite! The band should be between the two cysteines.
  - ***On good band:*** Great! Now wiggle to bring the cysteines together! If they don't form a bond, try alternating between shaking and wiggling, or lower the clashing importance and wiggling again.
    - ***On a disulfide bond:*** Excellent! Now practice on your own. There are two more pairs of cysteines that can form disulfide bonds in this puzzle.

Appendix H

Final Foldit Re-Design

This section describes the resulting tutorial design from Chapter 8. Levels are split into "Campaign" (progression-based challenges) and "Tutorial" (always available training). Tutorial levels are also embedded into the Campaign, such that Campaign levels require Tutorial levels which are accessible from the level selection menu.

H.1 World 1: Stabilize

This world focuses on introducing the player to the core mechanics. By the end of this set, players should be able to refine a protein structure.

Levels:

1. **Welcome** — *Tutorial* — Presents an overview of *Foldit*; why the player should play and broadly what is involved.
2. **Camera Controls** — *Tutorial* — Teaches how to control the camera.
3. **Wiggle** — *Tutorial* — Teaches how to use Wiggle, the most fundamental tool.
4. **Shake** — *Tutorial* — Teaches how to use Shake, the second most fundamental tool.
5. **Clashes and Voids** — *Tutorial* — Teaches the first principle of *Foldit* through two mechanical concepts: clashes represent parts of the protein that are too close, and voids represent parts of the protein that are too far apart.
6. **It's Unstable** — *Campaign* — The first challenge level; tests the player's understanding of everything learned so far (use Wiggle and Shake to remove Clashes and Voids).
7. **Clashing Importance** — *Tutorial* — Teaches a new tool, Clashing Importance, and introduces when and how to use it.

8. **Fuze** — *Campaign* — A challenge level, apply Clashing Importance to solve a textbook example of when this tool is useful.

H.2 World 2: Fold

This world focuses on the first major learning task: Prediction puzzles. It introduces several core tools needed for this task. By the end of this set, players should be able to solve Prediction puzzles.

Levels:

1. **Cut and Move** — *Tutorial* — Introduces two tools, Cut and Move, which are best used in combination.
2. **Hydrogen Bonds** — *Tutorial* — Teaches the second principle of *Foldit*, forming hydrogen bonds.
3. **Bands** — *Tutorial* — Introduces a new tool, (Rubber) Bands.
4. **Sheets Together** — *Campaign* — A challenge level, form hydrogen bonds using Bands and/or the 'Cut and Move' strategy.
5. **The Latest Issue** — *Campaign* — An optional challenge level, provides an increased challenge and additional practice on mechanics learned so far.
6. **One Step Forward...** — *Campaign* — An optional challenge level, provides an increased challenge and additional practice on mechanics learned so far.
7. **Idealize** — *Tutorial* — Teaches two new tools, Idealize and Ideal SS, which are used for similar purposes but disambiguated from each other.
8. **This Isn't Ideal** — *Campaign* — A challenge level which primarily tests the use of Idealize and Ideal SS.
9. **Prediction** — *Tutorial* — Teaches about the concept of Prediction puzzles as a scientific problem, although everything the player has been doing up to this point has been Prediction tasks.
10. **Prediction: Collagen** — *Campaign* — Simulates a real scientific Prediction puzzle. The goal is set to a challenging level of difficulty but full completion of this level is optional.

H.3 World 3: Design

This world focuses on the second major learning task: Design puzzles. It introduces several core tools needed for this task. By the end of this set, players should be able to solve Design puzzles.

Levels:

1. **Assigning Secondary Structures** — *Tutorial* — Teaches the player a new tool for assigning secondary structures.
2. **Spaghetti** — *Campaign* — A challenge level which tasks the player with practicing the new tool to assign secondary structures.
3. **Blueprint** — *Tutorial* — Teaches the player a new tool which makes assigning secondary structures easier and improves quality-of-life for the task overall.
4. **Uncooked Spaghetti** — *Campaign* — A more challenging version of the 'Spaghetti' puzzle; this level provides additional practice for applying the tools learned recently as well as practicing skills learned so far.
5. **Mutate** — *Tutorial* — Teaches the player a new tool for changing the amino acids of the protein.
6. **Orange In, Blue Out** — *Tutorial* — Introduces the third principle of *Foldit*, having hydrophobic (orange) amino acids on the inside of the protein and hydrophilic (blue) ones on the outside.
7. **Core Development** — *Campaign* — A challenge level; asks the player to use the new Mutate tool to make the protein 'orange in, blue out,' following the principle introduced in the last tutorial.
8. **Design** — *Tutorial* — Teaches about the concept of Design puzzles as a scientific problem, although most of the puzzles in this set have had elements of the Design task.
9. **Design: De Novo** — *Campaign* — Simulates a real scientific Design puzzle. The goal is set to a challenging level of difficulty but full completion of this level is optional.

Appendix I

Foldit Empirical Testing Details

This section provides additional detail on aspects of the empirical testing procedures in Chapter 8. Note that when assessing participant knowledge, different terms for the mechanics were accepted. For example, multiple participants referred to sheets as "plates" and bands as "ribbon," though why this is the case still eludes me. Similarly, "orange" and "blue" were accepted as substitutes for hydrophobic and hydrophilic.

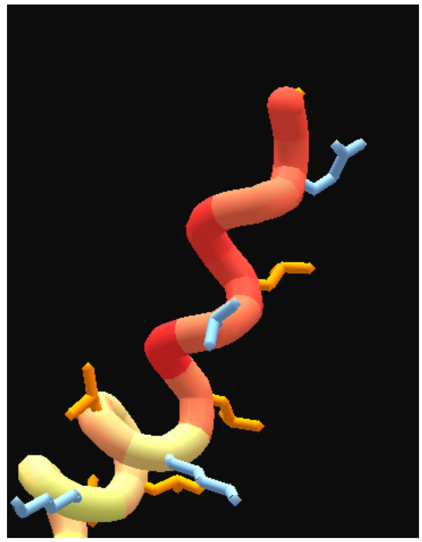
I.1 Phase 1 Quiz

Three questions were given in the Phase 1 quiz. Answers and accepted approximations of the answers are described in italics.

APPENDIX I. FOLDIT EMPIRICAL TESTING DETAILS

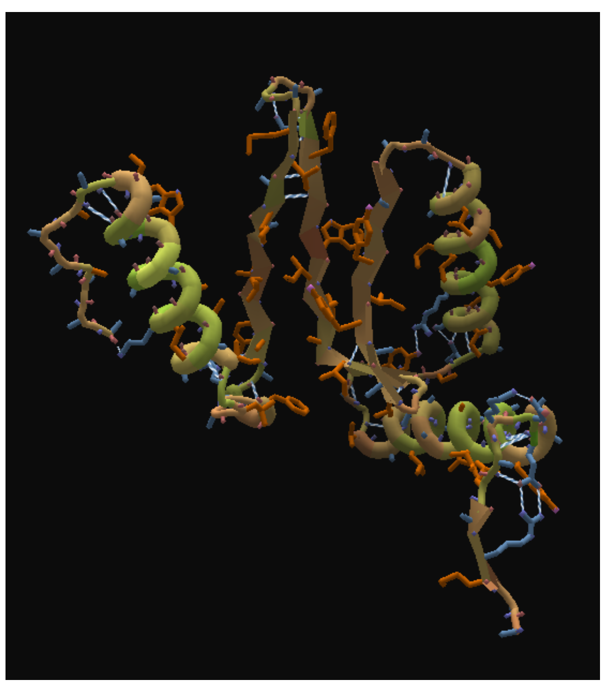
1. What tool should you use to improve this fold?

Idealize the secondary structure. Accepted: Idealize or Ideal SS.



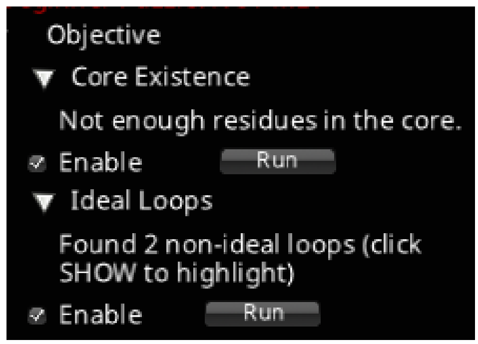
2. What would you change to improve this fold? (Include what tools you would use)

Bring the sheets together to form hydrogen bonds by using Rubber Bands or Pull. Accepted: bring the sheets together; form bonds; band the sheets.



3. Given these Objectives, how would you improve this fold? (Include what tools you would use)

Make ideal loops using the Blueprint tool or the Ramachandran Map and improve the core by placing hydrophobic residues inside and hydrophilic residues outside. Accepted: move orange inside; use Blueprint.

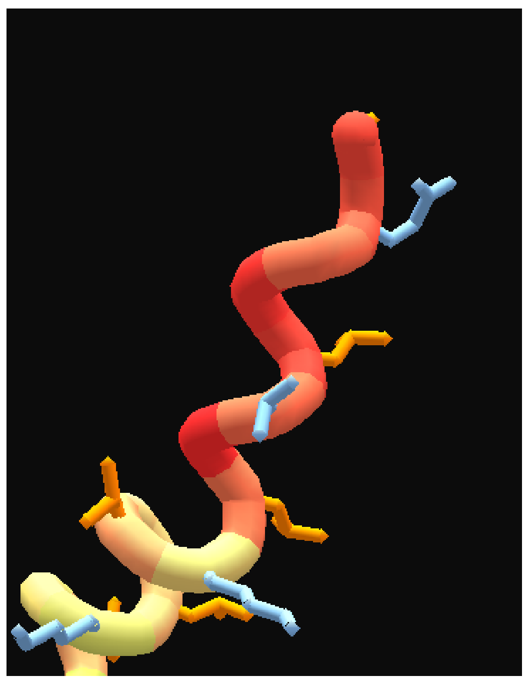


I.2 Phase 3 Quiz

Four questions were given in the Phase 3 quiz. Answers and accepted approximations of the answers are described in italics. Point values for each question are listed in bold.

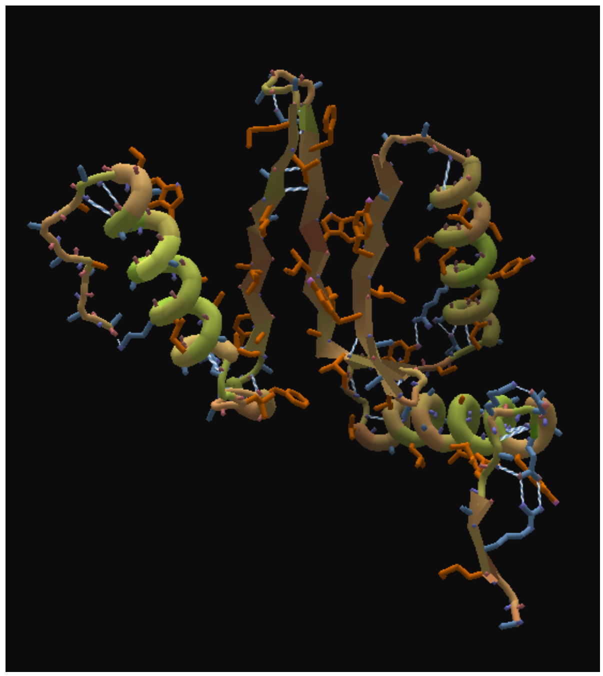
1. What tool should you use to improve this fold? You may refer to the game and hover over tools to inspect them. **2 points.** (NB*: This question was given extra weight because of how few participants correctly answered this question in Phase 1 and because the question tests information given near the end of the learning sequence, so additional weight gives value to participants who progressed further in their learning.)*

Idealize the secondary structure. Accepted: Idealize or Ideal SS.



2. How would you improve this fold? Which tool(s) would you use? You may refer to the game and hover over tools to inspect them. **1 point.**

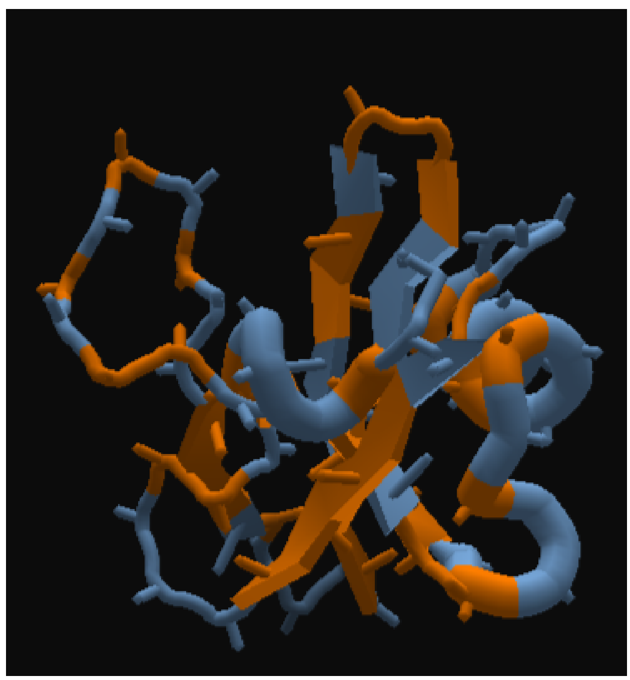
Bring the sheets together to form hydrogen bonds by using Rubber Bands or Pull. Accepted: bring the sheets together; form bonds; band the sheets.



3. What makes a protein stable? List as many factors as you can think of. **3 points.**

Close (avoid voids) but not too close (avoid clashes) — worth 0.5 points each, accepted "proper spacing" for 1 point; make hydrogen bonds (accepted any mention of bonds) — worth 1 point; hydrophobics on the inside and hydrophilics on the outside — worth 0.5 points each. Partial credit was given for actions that imply these principles, such as "bring sheets together" (implicitly forms bonds) or "bring backbones apart" (implies clashing).

4. What problems does this fold have? Name as many as you can and how you could fix them. You may refer to the game and hover over tools to inspect them. **3 points.**



Point values similar to Question 3 based on how the participant names problems and solutions using the principles of protein stability. For example, "Wiggle to remove clashes and voids" is worth 1 point. "Pull the orange inside" is worth 0.5 points.

Bibliography

[1] 3909 LLC. *Papers, Please*. Game [Windows, macOS, Linux, iOS, PlayStation Vita, Android], 2013. 3909 LLC.

[2] Espen Aarseth. Playing research: Methodological approaches to game analysis. In *Proceedings of the digital arts and culture conference*, pages 28–29, New York, NY, USA, 2003. RMIT University Melbourne, VIC, ACM.

[3] Rebecca Rashid Achterman. Minds at play: Using an online protein folding game, foldit, to support student learning about protein folding, structure, and the scientific process. *Journal of microbiology & biology education*, 20(3):100, 2019.

[4] Lada A Adamic. Zipf, power-laws, and pareto-a ranking tutorial. *Xerox Palo Alto Research Center, Palo Alto, CA*, 2000.

[5] Adamvision Studios. *Orbt XL*. Game [Windows, macOS, Linux, Android, iOS], 2017. Adamvision Studios.

[6] Adamvision Studios. *Ding Dong XL*. Game [Windows, macOS, Linux, Android, iOS], 2018. Adamvision Studios.

[7] Abdulmonem Alabri and Jane Hunter. Enhancing the quality and trust of citizen science data. In *2010 IEEE Sixth International Conference on e-Science*, pages 81–88. IEEE, 2010.

[8] Alan Hazelden and Lee Shang Lun. *Sokobond*. Game [Windows, macOS], 2014. Draknek.

[9] Vincent Aleven and Kenneth R Koedinger. Knowledge component (kc) approaches to learner modeling. *Design Recommendations for Intelligent Tutoring Systems*, 1:165–182, 2013.

[10] David Alpizar, Olusola O Adesope, and Rachel M Wong. A meta-analysis of signaling principle in multimedia learning environments. *Educational Technology Research and Development*, 68:2095–2119, 2020.

[11] Valter Alves and Licinio Roque. An inspection on a deck for sound design in games. In *Proceedings of the 6th Audio Mostly Conference: A Conference on Interaction with Sound*, pages 15–22, 2011.

[12] Erik Andersen, Yun-En Liu, Rich Snider, Roy Szeto, and Zoran Popović. Placing a value on aesthetics in online casual games. In *Proceedings of the SIGCHI Conference on human factors in Computing Systems*, pages 1275–1278, New York, NY, USA, 2011. Association for Computing Machinery.

[13] Erik Andersen, Yun-En Liu, Richard Snider, Roy Szeto, Seth Cooper, and Zoran Popović. On the harmfulness of secondary game objectives. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, pages 30–37. ACM, 2011.

[14] Erik Andersen, Eleanor O'Rourke, Yun-En Liu, Rich Snider, Jeff Lowdermilk, David Truong, Seth Cooper, and Zoran Popovic. The impact of tutorials on games of varying complexity. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 59–68, New York, NY, USA, 2012. ACM, ACM.

[15] Marc Malmdorf Andersen and Andreas Roepstorff. Play in predictive minds: A cognitive theory of play. *Psychological Review*, pages pre–print, 2022.

[16] John R Anderson, Lynne M Reder, and Herbert A Simon. Situated learning and education. *Educational researcher*, 25(4):5–11, 1996.

[17] Alissa N Antle, Theresa Jean Tanenbaum, Anna Macaranas, and John Robinson. Games for change: Looking at models of persuasion through the lens of design. In *Playful user interfaces*, pages 163–184. Springer, 2014.

[18] Aditya Anupam. Can digital games teach scientific inquiry? a case of how affordances can become constraints. *Proceedings of the ACM on Human-Computer Interaction*, 5(CHI PLAY):1–21, 2021.

[19] Louise Archer, Emily Dawson, Jennifer DeWitt, Amy Seakins, and Billy Wong. "science capital": A conceptual, methodological, and empirical argument for extending bourdieusian

notions of capital beyond the arts. *Journal of Research in Science Teaching*, 52(7):922–948, September 2015.

[20] Maria Aristeidou, Eileen Scanlon, and Mike Sharples. Weather-it: evolution of an online community for citizen inquiry. In *Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business*, pages 1–8, 2015.

[21] Maria Aristeidou, Eileen Scanlon, and Mike Sharples. Profiles of engagement in online communities of citizen science participation. *Computers in Human Behavior*, 74:246–256, 2017.

[22] Sylvester Arnab, Theodore Lim, Maira B Carvalho, Francesco Bellotti, Sara De Freitas, Sandy Louchart, Neil Suttie, Riccardo Berta, and Alessandro De Gloria. Mapping learning and game mechanics for serious games analysis. *British Journal of Educational Technology*, 46(2):391–411, 2015.

[23] ARTigo – Das Kunstgeschichtsspiel - Institut für Kunstgeschichte - LMU München.

[24] Kristine Ask. The value of calculations: The coproduction of theorycraft and player practices. *Bulletin of Science, Technology & Society*, 36(3):190–200, 2016.

[25] Sandeep Athavale and Girish Dalvi. Strategies for endogenous design of educational games. In *DiGRA Conference*, pages 1–19, Finland, 2019. DiGRA.

[26] Batu Aytemiz, Isaac Karth, Jesse Harder, Adam M Smith, and Jim Whitehead. Talin: A framework for dynamic tutorials based on the skill atoms theory. In *Fourteenth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2018.

[27] Marc Baaden, Olivier Delalande, Nicolas Ferey, Samuela Pasquali, Jérôme Waldispühl, and Antoine Taly. Ten simple rules to create a serious game, illustrated with examples from structural biology. *PLOS Computational Biology*, 14(3):e1005955, March 2018.

[28] Alan Baddeley. Working memory. *Science*, 255(5044):556–559, 1992.

[29] Claire Baert. Design approach in citizen science games, until eve online. *Gamasutra*. `https://www.gamasutra.com/blogs/ClaireBaert/20171017/307617/Design_approach_in_citizen_science_games_until_EVE_Online.php`, 2017.

[30] Sander CJ Bakkes, Pieter HM Spronck, and Giel van Lankveld. Player behavioural modelling for video games. *Entertainment Computing*, 3(3):71–79, 2012.

[31] Aaron Bangor, Philip T Kortum, and James T Miller. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction*, 24(6):574–594, 2008.

[32] David Bar-El and Kathryn E. Ringland. Crafting game-based learning: An analysis of lessons for minecraft education edition. In *International Conference on the Foundations of Digital Games*, pages 1–4, 2020.

[33] Christoph Bareither. Affordanz. In Timo Heimerdinger and Markus Tauschek, editors, *Kulturwissen-schaftlich Argumentieren. Ein Arbeitsbuch*, UTB, pages 32–55. Waxmann, münster/ new york edition, 2020.

[34] Yael Barel-Ben David, Erez S Garty, and Ayelet Baram-Tsabari. Can scientists fill the science journalism void? online public engagement with science stories authored by scientists. *PloS one*, 15(1):e0222250, 2020.

[35] Aaron Bauer and Zoran Popović. Collaborative problem solving in an open-ended scientific discovery game. *Proceedings of the ACM on human-computer interaction*, 1(CSCW):1–21, 2017.

[36] Roy F Baumeister and Mark R Leary. The need to belong: desire for interpersonal attachments as a fundamental human motivation. *Psychological bulletin*, 117(3):497, 1995.

[37] Roy F Baumeister, Jean M Twenge, and Christopher K Nuss. Effects of social exclusion on cognitive processes: anticipated aloneness reduces intelligent thought. *Journal of personality and social psychology*, 83(4):817, 2002.

[38] Katrin Becker. Instructional ethology: reverse engineering for serious design of educational games. In *Proceedings of the 2007 conference on Future Play*, pages 121–128, New York, NY, USA, 2007. ACM.

[39] Katrin Becker. *The invention of good games: understanding learning design in commercial video games*. PhD thesis, University of Calgary, 2008. Artwork Size: xxvi, 399 leaves : ill. ; 30 cm. Publisher: Graduate Studies.

[40] Nicolae Berbece. This is a talk about tutorials, press a to skip, 2019.

[41] Nicole A Beres, Julian Frommel, Elizabeth Reid, Regan L Mandryk, and Madison Klarkowski. Don't you know that you're toxic: Normalization of toxicity in online gaming. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2021.

[42] Joseph Bergin, Jutta Eckstein, Markus Volter, Marianna Sipos, Eugene Wallingford, Klaus Marquardt, Jane Chandler, Helen Sharp, and Mary Lynn Manns. *Pedagogical patterns: advice for educators*. Joseph Bergin Software Tools, 2012.

[43] Matthew L Bernacki and Candace Walkington. The role of situational interest in personalized learning. *Journal of Educational Psychology*, 110(6):864, 2018.

[44] Jim Bizzocchi and Theresa Jean Tanenbaum. Well read: Applying close reading techniques to gameplay experiences. *Well Played 3.0: Video Games, Value and Meaning*, 3:289–313, 2011.

[45] Adream Blair-Early and Mike Zender. User interface design principles for interaction design. *Design Issues*, 24(3):85–107, 2008.

[46] Lisa Marie Blaschke. *The pedagogy–andragogy–heutagogy continuum and technology-supported personal learning environments*, pages 75–84. Springer, Singapore, 2019.

[47] Blendo Games. *Air Forte*. Game [Windows, macOS], September 2010. Blendo Games.

[48] Matthew Bond and Russell Beale. What makes a good game?: using reviews to inform design. In *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology*, pages 418–422, New York, NY, USA, 2009. British Computer Society, Association for Computing Machinery.

[49] Elizabeth Bonsignore, Derek Hansen, Kari Kraus, and Marc Ruppel. Alternate reality games as platforms for practicing 21st-century literacies. *International Journal of Learning and Media*, 4(1):25–54, 2013.

[50] Geoffrey C. Bowker, Stefan Timmermans, Adele E. Clarke, and Ellen Balka, editors. *Boundary objects and beyond: working with Leigh Star*. Infrastructures. The MIT Press, Cambridge, Massachusetts, 2015.

[51] Anne Bowser, Derek Hansen, Yurong He, Carol Boston, Matthew Reid, Logan Gunnell, and Jennifer Preece. Using gamification to inspire new citizen science volunteers. In *Proceedings of the First International Conference on Gameful Design, Research, and Applications*, Gamification '13, page 18–25, New York, NY, USA, 2013. Association for Computing Machinery.

[52] Anne Bowser, Derek Hansen, and Jennifer Preece. Gamifying citizen science: Lessons and future directions. In *Workshop on Designing Gamification: Creating Gameful and Playful Experiences*, 2013.

[53] Anne Bowser, Derek Hansen, Jennifer Preece, Yurong He, Carol Boston, and Jen Hammock. Gamifying citizen science: a study of two user groups. In *Proceedings of the companion publication of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 137–140, 2014.

[54] Andy Boyan, Rory McGloin, and Joe A Wasserman. Model matching theory: A framework for examining the alignment between game mechanics and mental models. *Media and Communication*, 6(2):126–136, 2018.

[55] Kristy Elizabeth Boyer, Robert Phillips, Michael Wallis, Mladen Vouk, and James Lester. Balancing cognitive and motivational scaffolding in tutorial dialogue. In *International conference on intelligent tutoring systems*, pages 239–249, Cham, 2008. Springer, Springer.

[56] Alan Boyle. A protein puzzle game called foldit turns up 99 promising ways to confound coronavirus, 2020.

[57] Daren C Brabham. Crowdsourcing. massachusetts institute of technology, 2013.

[58] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2):77–101, 2006.

[59] Virginia Braun and Victoria Clarke. Reflecting on reflexive thematic analysis. *Qualitative Research in Sport, Exercise and Health*, 11(4):589–597, 2019.

[60] Virginia Braun and Victoria Clarke. To saturate or not to saturate? questioning data saturation as a useful concept for thematic analysis and sample-size rationales. *Qualitative Research in Sport, Exercise and Health*, pages 1–16, 2019.

[61] Virginia Braun and Victoria Clarke. Can i use ta? should i use ta? should i not use ta? comparing reflexive thematic analysis and other pattern-based qualitative analytic approaches. *Counselling and Psychotherapy Research*, 21(1):37–47, 2021.

[62] John Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[63] Ann L Brown and Joseph C Campione. *Guided discovery in a community of learners.* The MIT Press, 1994.

[64] Emily Brown and Paul Cairns. A grounded investigation of game immersion. In *CHI'04 extended abstracts on Human factors in computing systems*, pages 1297–1300. ACM, 2004.

[65] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[66] Olivia M Bullock, Daniel Colón Amill, Hillary C Shulman, and Graham N Dixon. Jargon as a barrier to effective science communication: Evidence from metacognition. *Public Understanding of Science*, 28(7):845–853, 2019.

[67] Regula Valérie Burri. Models of Public Engagement: Nanoscientists' Understandings of Science–Society Interactions. *NanoEthics*, 12(2):81–98, August 2018.

[68] Gordon Calleja. *In-game: From immersion to incorporation.* mit Press, 2011.

[69] Elin Carstensdottir, Erica Kleinman, and Magy Seif El-Nasr. Player interaction in narrative games: structure and narrative progression mechanics. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, pages 1–9, New York, NY, USA, 2019. ACM.

[70] Luigi Ceccaroni, Anne Bowser, and Peter Brenton. Civic education and citizen science: Definitions, categories, knowledge representation. In *Analyzing the role of citizen science in modern research*, pages 1–23. IGI Global, 2017.

[71] Nicholas J Cepeda, Harold Pashler, Edward Vul, John T Wixted, and Doug Rohrer. Distributed practice in verbal recall tasks: A review and quantitative synthesis. *Psychological bulletin*, 132(3):354, 2006.

[72] Kathy Charmaz. *Constructing grounded theory*. Introducing qualitative methods. Sage, London ; Thousand Oaks, Calif, 2nd edition edition, 2014. OCLC: ocn878133162.

[73] William G. Chase and Herbert A. Simon. Perception in chess. *Cognitive psychology*, 4(1):55–81, 1973.

[74] Jenova Chen. Flow in games (and everything else). *Communications of the ACM*, 50(4):31–34, 2007.

[75] Jilin Chen, Yuqing Ren, and John Riedl. The effects of diversity on group productivity and member withdrawal in online volunteer groups. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 821–830, 2010.

[76] Meng-Tzu Cheng, Louisa Rosenheck, Chen-Yen Lin, and Eric Klopfer. Analyzing gameplay data to inform feedback loops in the radix endeavor. *Computers & Education*, 111:60–73, 2017.

[77] Gifford K Cheung, Thomas Zimmermann, and Nachiappan Nagappan. The first hour experience: how the initial play can engage (or lose) new players. In *Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play*, pages 57–66, New York, NY, USA, 2014. ACM.

[78] Michelene TH Chi, Paul J Feltovich, and Robert Glaser. Categorization and representation of physics problems by experts and novices. *Cognitive science*, 5(2):121–152, 1981.

[79] Hichang Cho, MeiHui Chen, and Siyoung Chung. Testing an integrative theoretical model of knowledge-sharing behavior in the context of wikipedia. *Journal of the American Society for Information Science and Technology*, 61(6):1198–1212, 2010.

[80] Chucklefish. *Starbound*. Game [Windows, macOS, Linux], 2016. Chucklefish.

[81] Citizen Sort. *Forgotten Island*. Game [Web Browser], 2013. `https://citizensort.org/web.php/forgottenisland`.

[82] James M Clark and Allan Paivio. Dual coding theory and education. *Educational psychology review*, 3(3):149–210, 1991.

[83] Oscar Clark. *Games as a service: How free to play design can make better games*. Routledge, 2014.

[84] Richard Clark. Cognitive task analysis for expert-based instruction in healthcare. In *Handbook of research on educational communications and technology*, pages 541–551. Springer, 2014.

[85] Richard Clark, Paul A Kirschner, and John Sweller. Putting students on the path to learning: The case for fully guided instruction. *American Educator*, 36(1):5–11, 2012.

[86] Richard E Clark, Carla M Pugh, Kenneth A Yates, Kenji Inaba, Donald J Green, and Maura E Sullivan. The use of cognitive task analysis to improve instructional descriptions of procedures. *Journal of Surgical Research*, 173(1):e37–e42, 2012.

[87] Richard E Clark, Kenneth Yates, Sean Early, and Kathrine Moulton. An analysis of the failure of electronic media and discovery-based learning: Evidence for the performance benefits of guided training methods. *Handbook of Improving Performance in the Workplace: Volumes 1-3*, pages 263–297, 2009.

[88] Claudiu Kiss and The Irregular Corporation. *PC Building Simulator*. Game [Windows, PlayStation 4, Xbox One, Nintendo Switch], January 2019. The Irregular Corporation.

[89] Cassandra Coburn. Play to cure: Genes in space. *The Lancet Oncology*, 15(7):688, 2014.

[90] Coldwild Games, Vladimir Slav, and Elena Nazarenko. *Lazy Galaxy*. Game [Windows, macOS, Linux], 2018. Coldwild Games.

[91] Allan Collins. Cognitive apprenticeship and instructional technology. *Educational values and cognitive instruction: Implications for reform*, 1991:121–138, 1991.

[92] Allan Collins, John Seely Brown, and Ann Holum. Cognitive apprenticeship: Making thinking visible. *American educator*, 15(3):6–11, 1991.

[93] Allan Collins, John Seely Brown, and Susan E Newman. Cognitive apprenticeship: Teaching the craft of reading, writing and mathematics. *Thinking: The Journal of Philosophy for Children*, 8(1):2–10, 1988.

[94] Mia Consalvo. When paratexts become texts: de-centering the game-as-text. *Critical Studies in Media Communication*, 34(2):177–183, 2017.

[95] Daniel Cook. The chemistry of game design. *Gamasutra*. `http://www.gamasutra.com/view/feature/1524/the_chemistry_of_game_design.php`, 2007.

[96] Nancy J Cooke. Knowledge elicitation. *Handbook of applied cognition*, pages 479–509, 1999.

[97] Seth Cooper. *A framework for scientific discovery through video games*. Morgan & Claypool, 2014.

[98] Seth Cooper, Firas Khatib, Ilya Makedon, Hao Lu, Janos Barbero, David Baker, James Fogarty, Zoran Popović, and Foldit Players. Analysis of social gameplay macros in the foldit cookbook. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, pages 9–14, 2011.

[99] Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popović, et al. Predicting protein structures with a multiplayer online game. *Nature*, 466(7307):756, August 2010.

[100] Seth Cooper, Adrien Treuille, Janos Barbero, Andrew Leaver-Fay, Kathleen Tuite, Firas Khatib, Alex Cho Snyder, Michael Beenen, David Salesin, David Baker, et al. The challenge of designing scientific discovery games. In *Proceedings of the Fifth international Conference on the Foundations of Digital Games*, pages 40–47, 2010.

[101] Ben Cowley, Darryl Charles, Michaela Black, and Ray Hickey. Toward an understanding of flow in video games. *Computers in Entertainment (CIE)*, 6(2):20, 2008.

[102] Benjamin Cowley and Darryl Charles. Behavlets: a method for practical player modelling using psychology-based player traits and domain specific features. *User Modeling and User-Adapted Interaction*, 26(2-3):257–306, 2016.

[103] Beth Crandall, Gary A. Klein, and Robert R. Hoffman. *Working minds: a practitioner's guide to cognitive task analysis*. MIT Press, Cambridge, Mass, 2006. OCLC: ocm62393256.

[104] Justin Cranshaw and Aniket Kittur. The polymath project: lessons from a successful online collaboration in mathematics. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1865–1874, 2011.

[105] John W. Creswell and Dana L. Miller. Determining validity in qualitative inquiry. *Theory Into Practice*, 39(3):124–130, August 2000.

[106] David Crookall. Serious Games, Debriefing, and Simulation/Gaming as a Discipline. *Simulation & Gaming*, 41(6):898–920, December 2010.

[107] Kevin Crowston and Nathan R Prestopnik. Motivation and data quality in a citizen science game: A design science evaluation. In *2013 46th Hawaii International Conference on System Sciences*, pages 450–459. IEEE, 2013.

[108] Christian Crumlish and Erin Malone. *Designing social interfaces: Principles, patterns, and practices for improving the user experience*. " O'Reilly Media, Inc.", 2009.

[109] Mihaly Csikszentmihalyi. Flow: The psychology of optimal performance. *NY: Cambridge UniversityPress*, 40, 1990.

[110] Mihaly Csikszentmihalyi. Flow and the psychology of discovery and invention. *Harper-Perennial, New York*, 39, 1997.

[111] Mihaly Csikszentmihalyi and Isabella Csikszentmihalyi. *Beyond boredom and anxiety*, volume 721. Jossey-Bass San Francisco, 1975.

[112] Sabrina Culyba. *The Transformational Framework: A process tool for the development of Transformational games*. Carnegie Mellon University, 2018.

[113] Tom Curtis. GDC 2012: 10 tutorial tips from plants vs. zombies creator george fan, March 2012.

[114] Vickie Curtis. Motivation to Participate in an Online Citizen Science Game: A Study of Foldit. *Science Communication*, 37(6):723–746, December 2015.

[115] Vickie Curtis. Who takes part in online citizen science? In *Online Citizen Science and the Widening of Academia*, pages 45–68. Springer, Cham, 2018.

[116] Joe Cutting and Sebastian Deterding. The task-attention theory of game learning: a theory and research agenda. *Human–Computer Interaction*, pages 1–31, 2022.

[117] Fred D Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, pages 319–340, 1989.

[118] Jan W de Fockert, Geraint Rees, Christopher D Frith, and Nilli Lavie. The role of working memory in visual selective attention. *Science*, 291(5509):1803–1806, 2001.

[119] Adriaan de Jongh. Playtesting: Avoiding Evil Data, 2017.

[120] MJ de Vries, AM Land-Zandstra, and Ionica Smeets. Citizen scientists' preferences for communication of scientific output: a literature review. *Citizen Science: Theory and Practice*, 4(1):2, 2019.

[121] Edward L Deci and Richard M Ryan. Cognitive evaluation theory. In *Intrinsic motivation and self-determination in human behavior*, pages 43–85. Springer, 1985.

[122] Edward L Deci and Richard M Ryan. The "what" and "why" of goal pursuits: Human needs and the self-determination of behavior. *Psychological inquiry*, 11(4):227–268, 2000.

[123] Edward L Deci and Richard M Ryan. Self-determination theory: A macrotheory of human motivation, development, and health. *Canadian psychology/Psychologie canadienne*, 49(3):182, 2008.

[124] Juan Carlos Sandí Delgado and Patricia Bazán. Educational serious games as a service: Challenges and solutions. *Journal of Computer Science and Technology*, 19(01):e07–e07, 2019.

[125] André R. Denham. Improving the Design of a Learning Game Through Intrinsic Integration and Playtesting. *Technology, Knowledge and Learning*, 21(2):175–194, July 2016.

[126] Michel C. Desmarais and Ryan S.J.d. Baker. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, 22(1-2):9–38, 2012.

[127] Heather Desurvire, Martin Caplan, and Jozsef A Toth. Using heuristics to evaluate the playability of games. In *CHI'04 extended abstracts on Human factors in computing systems*, pages 1509–1512, New York, NY, USA, 2004. ACM, ACM.

[128] Heather Desurvire and Charlotte Wiberg. Master of the game: Assessing approachability in future game design. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '08, pages 3177–3182, New York, NY, USA, 2008. ACM.

[129] Heather Desurvire and Charlotte Wiberg. Game usability heuristics (play) for evaluating and designing better games: The next iteration. In *International Conference on Online Communities and Social Computing*, pages 557–566. Springer, 2009.

[130] Heather Desurvire and Charlotte Wiberg. User experience design for inexperienced gamers: Gap—game approachability principles. In *Game User Experience Evaluation*, pages 169–186. Springer, Cham, 2015.

[131] Sebastian Deterding. Gamification: designing for motivation. *interactions*, 19(4):14–17, 2012.

[132] Sebastian Deterding. *Modes of play: A frame analytic account of video game play*. PhD thesis, Staats-und Universitätsbibliothek Hamburg Carl von Ossietzky, 2013.

[133] Sebastian Deterding. The ambiguity of games: Histories and discourses of a gameful world. *SP Walz, & S. Deterding, The Gameful World: Approaches, Issues, Applications*, pages 23–64, 2014.

[134] Sebastian Deterding. The lens of intrinsic skill atoms: A method for gameful design. *Human–Computer Interaction*, 30(3-4):294–335, 2015.

[135] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. From game design elements to gamefulness: defining" gamification". In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, pages 9–15, 2011.

[136] Carlos Díaz, Marisa Ponti, Pinja Haikka, Rajiv Basaiawmoit, and Jacob Sherson. More than data gatherers: exploring player experience in a citizen science game. *Quality and User Experience*, 5(1):1, 2020.

[137] Darina Dicheva, Christo Dichev, Gennady Agre, and Galia Angelova. Gamification in education: A systematic mapping study. *Journal of educational technology & society*, 18(3):75–88, 2015.

[138] Dinosaur Polo Club. *Mini Metro*. Game [Windows, macOS, Linux, Android, iOS], 2015. Dinosaur Polo Club.

[139] Anne Dippel and Sonia Fizek. Laborious playgrounds: Citizen science games as new modes of work/play in the digital age. In *The playful citizen: civic engagement in a mediatized culture*. Amsterdam University Press, 2018.

[140] Iain Donald, Karen A Meyer, John Brengman, Stephen H Gillespie, and Ruth Bowness. Project sanitarium: playing tuberculosis to its end game. *Journal of Computing in Higher Education*, 29:599–617, 2017.

[141] Scott Donaldson. Mechanics and metagame: Exploring binary expertise in league of legends. *Games and Culture*, 12(5):426–444, 2017.

[142] Anders Drachen, Pejman Mirza-Babaei, and Lennart E Nacke. *Games user research*. Oxford University Press, 2018.

[143] Dry Cactus. *Poly Bridge*. Game [Windows, macOS, Linux, Android, iOS, Nintendo Switch], 2016. Dry Cactus.

[144] Karl Duncker and Lynne S Lees. On problem-solving. *Psychological monographs*, 58(5):i, 1945.

[145] Carlos Díaz, Marisa Ponti, Pinja Haikka, Rajiv Basaiawmoit, and Jacob Sherson. More than data gatherers: exploring player experience in a citizen science game. *Qual User Exp*, 5(1):1, December 2020.

[146] Peter Eastman, Jade Shi, Bharath Ramsundar, and Vijay S Pande. Solving the rna design problem with reinforcement learning. *PLoS computational biology*, 14(6):e1006176, 2018.

[147] Hermann Ebbinghaus. Memory: A contribution to experimental psychology. *Annals of neurosciences*, 20(4):155, 2013.

[148] Melissa V Eitzel, Jessica L Cappadonna, Chris Santos-Lang, Ruth Ellen Duerr, Arika Virapongse, Sarah Elizabeth West, Christopher Kyba, Anne Bowser, Caren Beth Cooper, Andrea Sforzi, et al. Citizen science terminology matters: Exploring key terms. *Citizen science: Theory and practice*, 2(1):1, June 2017.

[149] Magy Seif El-Nasr, Shree Durga, Mariya Shiyko, and Carmen Sceppa. Unpacking adherence and engagement in pervasive health games. In *Proceedings of the 10th International Conference on Foundations of Digital Games*, New York, NY, USA, 2015. Association for Computing Machinery.

[150] Satu Elo and Helvi Kyngäs. The qualitative content analysis process. *Journal of advanced nursing*, 62(1):107–115, 2008.

[151] Séverine Erhel and Eric Jamet. Digital game-based learning: Impact of instructions and feedback on motivation and learning effectiveness. *Computers & education*, 67:156–167, 2013.

[152] Eterna players. Player-created guides, 2021. Retrieved from `http://eternawiki.org/wiki/index.php5/Player-Created_Guides` on November 29, 2021.

[153] Alexandra Eveleigh, Charlene Jennett, Ann Blandford, Philip Brohan, and Anna L Cox. Designing for dabblers and deterring drop-outs in citizen science. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2985–2994, New York, NY, USA, 2014. ACM, Association for Computing Machinery.

[154] Alexandra Eveleigh, Charlene Jennett, Stuart Lynn, and Anna L Cox. "i want to be a captain! i want to be a captain!" gamification in the old weather citizen science project. In *Proceedings of the first international conference on gameful design, research, and applications*, pages 79–82, 2013.

[155] Extremely OK Games, Ltd. *TowerFall Ascension*. Game [Windows, macOS, Linux], 2018. Matt Makes Games Inc.

[156] Michael W Eysenck, Nazanin Derakshan, Rita Santos, and Manuel G Calvo. Anxiety and cognitive performance: attentional control theory. *Emotion*, 7(2):336, 2007.

[157] Travis Faas, Lynn Dombrowski, Alyson Young, and Andrew D Miller. Watch me code: Programming mentorship communities on twitch. tv. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):1–18, 2018.

[158] Joey R Fanfarelli. Expertise in professional overwatch play. *International Journal of Gaming and Computer-Mediated Simulations (IJGCMS)*, 10(1):1–22, 2018.

[159] Peter C Farley. Using the computer game "foldit" to entice students to explore external representations of protein structure in a biochemistry course for nonmajors. *Biochemistry and Molecular Biology Education*, 41(1):56–57, 2013.

[160] David Farrell and David Moffat. Applying the self determination theory of motivation in games based learning. In *European Conference on Games Based Learning*, volume 1, page 118. Academic Conferences International Limited, 2014.

[161] Melissa A Federoff. *Heuristics and usability guidelines for the creation and evaluation of fun in video games*. PhD thesis, Citeseer, 2002.

[162] Fireproof Games. *The Room Three*. Game [Windows, Android, iOS], 2015. Fireproof Games.

[163] Jane Forman and Laura Damschroder. Qualitative content analysis. In *Empirical methods for bioethics: A primer*. Emerald Group Publishing Limited, 2007.

[164] Lucy Fortson, Karen Masters, Robert Nichol, EM Edmondson, C Lintott, J Raddick, and J Wallin. Galaxy zoo. *Advances in machine learning and data mining for astronomy*, 2012:213–236, 2012.

[165] Judy Foster-Smith and Stewart M Evans. The value of marine ecological data collected by volunteers. *Biological Conservation*, 113(2):199–213, 2003.

[166] Anders Frank. Balancing three different foci in the design of serious games: Engagement, training objective and context. In *DiGRA Conference*, 2007.

[167] Benjamin Franklin. *Experiments and observations on electricity, made at Philadelphia in America*. David Henry, 1769.

[168] Chiara Franzoni and Henry Sauermann. Crowd science: The organization of scientific research in open collaborative projects. *Research policy*, 43(1):1–20, 2014.

[169] Julian Frommel, Kim Fahlbusch, Julia Brich, and Michael Weber. The effects of context-sensitive tutorials in virtual reality games. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, pages 367–375, New York, NY, USA, 2017. ACM.

[170] Zach Gage. Designing for problem-solvers. PRACTICE: Game Design in Detail, 2014. Retrieved from http://stfj.net/DesigningForProblem-Solvers/ on May 13, 2021.

[171] Kutub Gandhi, Josh Aaron Miller, and Seth Cooper. "philosophy is seeped into every brick"-weaving reflective elements into mass-market games. In *Proceedings of the 17th International Conference on the Foundations of Digital Games*, pages 1–10, New York, NY, USA, 2022. ACM.

[172] Barry Garelick. Problem solving: Moving from routine to nonroutine and beyond. *Notices of the AMs*, 60(10), 2013.

[173] Jacqueline Gaston and Seth Cooper. To three or not to three: Improving human computation game onboarding with a three-star system. In *Proceedings of the 2017 CHI conference on Human Factors in Computing Systems*, pages 5034–5039, New York, NY, USA, 2017. ACM.

[174] Matthew J Gaydos. *Design-based research and video game based learning: Developing the educational video game" Citizen Science"*. PhD thesis, The University of Wisconsin-Madison, 2013.

[175] James Paul Gee. What video games have to teach us about learning and literacy. *Computers in Entertainment (CIE)*, 1(1):20–20, 2003.

[176] James Paul Gee. *An introduction to discourse analysis: Theory and method*. Routledge, 2004.

[177] James Paul Gee. Good video games and good learning. *Phi Kappa Phi Forum*, 85(2):33, 2005.

[178] James Paul Gee. Learning by design: Good video games as learning machines. *E-learning and Digital Media*, 2(1):5–16, 2005.

[179] James Paul Gee. Semiotic social spaces and affinity spaces. *Beyond communities of practice language power and social context*, 214232, 2005.

[180] James Paul Gee. *Good video games+ good learning: Collected essays on video games, learning, and literacy*. Peter Lang, Switzerland, 2007.

[181] James Paul Gee. Discourse, small d, big d. *The international encyclopedia of language and social interaction*, 3:1–5, 2015.

[182] CHRIS Georgen. Well played & well watched: Dota 2, spectatorship, and esports. *Well Played A Journal on Video Games, Values, and Meaning*, 4(1):179–191, 2015.

[183] James J. Gibson. *The Ecological Approach to Visual Perception: Classic Edition*. Psychology Press, New York, 1979.

[184] Glitch Games. *Veritas*. Game [Windows, macOS], 2020. Glitch Games.

[185] Aytac Gogus. Peer Learning and Assessment. In Norbert M. Seel, editor, *Encyclopedia of the Sciences of Learning*, pages 2572–2576. Springer US, Boston, MA, 2012.

[186] Yaela N. Golumbic, Daniela Orr, Ayelet Baram-Tsabari, and Barak Fishbain. Between Vision and Reality: A Study of Scientists' Views on Citizen Science. *Citizen Science: Theory and Practice*, 2(1):6, October 2017.

[187] Carolina González, Juan C Burguillo, and Martín Llamas. A qualitative comparison of techniques for student modeling in intelligent tutoring systems. In *Proceedings. Frontiers in Education. 36th Annual Conference*, pages 13–18. IEEE, 2006.

[188] Charles Goodwin. Professional vision. *American Anthropologist*, 96(3):606, 1994.

[189] Charles Goodwin. Professional vision. In *Aufmerksamkeit*, pages 387–425. Springer, 2015.

[190] Kris Graft. Designing shadow of mordor's nemesis system. *Gamasutra*. `http://www.gamasutra.com/view/news/235777/Designing_Shadow_of_Mordors_Nemesis_system.php`, 2015.

[191] Tiago de Araujo Guerra Grangeia, Bruno de Jorge, Daniel Franci, Thiago Martins Santos, Maria Silvia Vellutini Setubal, Marcelo Schweller, and Marco Antonio de Carvalho-Filho. Cognitive load and self-determination theories applied to e-learning: impact on students' participation and academic performance. *PloS one*, 11(3):e0152462, 2016.

[192] Kevin Greenberg and Robert Zheng. Revisiting the debate on germane cognitive load versus germane resources. *Journal of Cognitive Psychology*, 0(0):1–20, 2022.

[193] Greenheart Games. *Game Dev Tycoon*. Game [Windows, macOS, Linux, iOS, Android, Nintendo Switch], 2012. Greenheart Games.

[194] Anita Greenhill, Kate Holmes, Chris Lintott, Brooke Simmons, Karen Masters, Joe Cox, and Gary Graham. Playing with science: Gamised aspects of gamification found on the online citizen science project-zooniverse. In *GAMEON'2014*, pages 15–24. EUROSIS, 2014.

[195] Anita Greenhill, Kate Holmes, Jamie Woodcock, Chris Lintott, Brooke D Simmons, Gary Graham, Joe Cox, Eun Young Oh, and Karen Masters. Playing with science. *Aslib Journal of Information Management*, 2016.

[196] Tovi Grossman, George Fitzmaurice, and Ramtin Attar. A survey of software learnability: metrics, methodologies and guidelines. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 649–658, 2009.

[197] Greg Guest, Arwen Bunce, and Laura Johnson. How many interviews are enough? an experiment with data saturation and variability. *Field methods*, 18(1):59–82, 2006.

[198] Greg Guest and Kathleen M MacQueen. *Handbook for team-based qualitative research*. Rowman Altamira, 2008.

[199] h-bar. *meQuanics*. Game [Web Browser], 2016. `http://www.mequanics.com.au/`.

[200] MP Jacob Habgood and Shaaron E Ainsworth. Motivating children to learn effectively: Exploring the value of intrinsic integration in educational games. *The Journal of the Learning Sciences*, 20(2):169–206, 2011.

[201] Thomas Hainey, Thomas M. Connolly, Elizabeth A. Boyle, Amanda Wilson, and Aisya Razak. A systematic literature review of games-based learning empirical evidence in primary education. *Computers & Education*, 102:202–223, November 2016.

[202] Muki Haklay. Citizen Science and Volunteered Geographic Information: Overview and Typology of Participation. In Daniel Sui, Sarah Elwood, and Michael Goodchild, editors, *Crowdsourcing Geographic Knowledge*, pages 105–122. Springer Netherlands, Dordrecht, 2013.

[203] Richard Halverson. What can k-12 school leaders learn from video games and gaming? *Innovate: journal of online education*, 1(6):3, 2005.

[204] Allison W Harrison, R Kelly Rainer Jr, Wayne A Hochwarter, and Kenneth R Thompson. Testing the self-efficacy—performance linkage of social—cognitive theory. *The Journal of social psychology*, 137(1):79–87, 1997.

[205] Sandra G Hart. Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, pages 904–908, Los Angeles, CA, 2006. Sage Publications Sage CA: Los Angeles, CA, Sage Publications.

[206] Casper Harteveld. *Triadic game design: Balancing reality, meaning and play*. Springer Science & Business Media, 2011.

[207] Joshua K Hartshorne, Claire Bonial, and Martha Palmer. The verbcorner project: Findings from phase 1 of crowd-sourcing a semantic decomposition of verbs. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 397–402, 2014.

[208] Anja Hawlitschek and Sven Joeckel. Increasing the effectiveness of digital educational games: The effects of a learning instruction on students' learning, motivation and cognitive load. *Computers in Human Behavior*, 72:79–86, 2017.

[209] Caroline Haythornthwaite. Crowds and communities: Light and heavyweight models of peer production. In *2009 42nd Hawaii international conference on system sciences*, pages 1–10. IEEE, 2009.

[210] Nathan Hedges. Video game tutorials: How do they teach?, Oct 2017.

[211] Florian Heigl, Barbara Kieslinger, Katharina T Paul, Julia Uhlik, and Daniel Dörler. Opinion: Toward an international definition of citizen science. *Proceedings of the National Academy of Sciences*, 116(17):8089–8092, 2019.

[212] Hempuli. *Baba Is You*. Game [Windows, macOS, Linux, Nintendo Switch], 2019. Hempuli.

[213] Joseph Hesketh, Christoph Sebastian Deterding, and Jeremy Gow. How players learn team-versus-team esports: First results from a grounded theory study. In *DiGRA'20 Abstract-Proceedings of the 2020 DiGRA International Conference*. York, 2020.

[214] Per Hetland, Palmyre Pierroux, and Line Esborg. *A history of participation in museums and archives: Traversing citizen science and citizen humanities*. Taylor & Francis, 2020.

[215] Celia Hodent. *The gamer's brain: How neuroscience and UX can impact video game design*. CRC Press, Boca Raton, Florida, USA, 2017.

[216] Robert R. Hoffman, John W. Coffey, Mary Jo Carnot, and Joseph D. Novak. An empirical comparison of methods for eliciting and modeling expert knowledge. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, pages 482–486, Los Angeles, CA, 2002. SAGE Publications Sage CA: Los Angeles, CA, SAGE Publications.

[217] Britton Horn. *A Skill-based Framework to Analyze the Generation and Presentation of Instructional Videogame Content*. PhD thesis, Northeastern University, 2019.

[218] Britton Horn, Seth Cooper, and Sebastian Deterding. Adapting cognitive task analysis to elicit the skill chain of a game. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, pages 277–289, 2017.

[219] Britton Horn, Josh Aaron Miller, Gillian Smith, and Seth Cooper. A monte carlo approach to skill-based automated playtesting. In *Fourteenth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2018.

[220] Jeff Howe et al. The rise of crowdsourcing. *Wired magazine*, 14(6):1–4, 2006.

[221] Hsiu-Fang Hsieh and Sarah E Shannon. Three approaches to qualitative content analysis. *Qualitative health research*, 15(9):1277–1288, 2005.

[222] Yuling Hsu, Yuan Gao, Tzu-Chien Liu, and John Sweller. Interactions between levels of instructional detail and expertise when learning with computer simulations. *Educational Technology & Society*, 18(4):113–127, 2015.

[223] Jeff Huang, Eddie Yan, Gifford Cheung, Nachiappan Nagappan, and Thomas Zimmermann. Master maker: Understanding gaming skill through practice and habit from gameplay behavior. *Topics in cognitive science*, 9(2):437–466, 2017.

[224] Wenhao David Huang and Tristan Johnson. Instructional game design using cognitive load theory. In *Instructional Design: Concepts, Methodologies, Tools and Applications*, pages 1586–1606. IGI Global, 2011.

[225] Robin Hunicke, Marc LeBlanc, and Robert Zubek. Mda: A formal approach to game design and game research. In *Proceedings of the AAAI Workshop on Challenges in Game AI*, pages 1–5, Washington, DC, USA, 2004. AAAI Press.

[226] Matthew Hyde, Maxwell Scott-Slade, Hugo Scott-Slade, Michael Hornberger, Hugo Spiers, Ruth Dalton, Christoph Hoelscher, Jan Wiener, and Veronique Bohbot. Sea hero quest: the world's first mobile game where anyone can help scientists fight dementia, 2016. Lancaster University. https://eprints.lancs.ac.uk/id/eprint/138622/.

[227] Ioanna Iacovides, James C Aczel, Eileen Scanlon, and Will IS Woods. What can breakdowns and breakthroughs tell us about learning and involvement experienced during game-play. In *Proceedings of the 5th European Conference on Games Based Learning*, pages 275–281, Athens, Greece, 2011. Academic Publishing International Reading, UK, Academic Conferences Ltd.

[228] Ioanna Iacovides, Anna L Cox, Patrick McAndrew, James Aczel, and Eileen Scanlon. Game-play breakdowns and breakthroughs: exploring the relationship between action, understanding, and involvement. *Human–computer interaction*, 30(3-4):202–231, 2015.

[229] Ioanna Iacovides, Charlene Jennett, Cassandra Cornish-Trestrail, and Anna L Cox. Do games attract or sustain engagement in citizen science?: a study of volunteer motivations. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, pages 1101–1106, New York, NY, USA, 2013. ACM, ACM.

[230] Ioanna Iacovides, Patrick McAndrew, Eileen Scanlon, and James Aczel. The gaming involvement and informal learning framework. *Simulation & gaming*, 45(4-5):611–626, 2014.

[231] Ian MacLarty and Jon Kerney. *Boson X*. Game [Windows, macOS, Linux], 2014. Shape Shop.

[232] Immersed Games. *Tyto Ecology*. Game [Windows, macOS], 2016. Immersed Games.

[233] inaturalist, 2021. Retrieved from `https://inaturalist.org/` on December 2, 2021.

[234] Katherine Isbister. *How games move us: Emotion by design*. Mit Press, 2016.

[235] Mizuko Ito, Kris Gutiérrez, Sonia Livingstone, Bill Penuel, Jean Rhodes, Katie Salen, Juliet Schor, Julian Sefton-Green, and S Craig Watkins. *Connected learning: An agenda for research and design*. Digital Media and Learning Research Hub, 2013.

[236] It's happening. *PlateUp!* Game [Windows], 2022. Yogscast Games.

[237] Aki Jarvinen. First five minutes: How tutorials make or break your social game. *Gamasutra*. `https://www.gamasutra.com/view/feature/132715/first_five_minutes_how_tutorials_.php?`, 2010.

[238] Chaima Jemmali, Sara Bunian, Andrea Mambretti, and Magy Seif El-Nasr. Educational game design: an empirical study of the effects of narrative. In *Proceedings of the 13th international conference on the foundations of digital games*, pages 1–10, New York, NY, USA, 2018. ACM.

[239] Charlene Jennett, Dominic J Furniss, Ioanna Iacovides, Sarah Wiseman, Sandy JJ Gould, and Anna L Cox. Exploring citizen psych-science and the motivations of errordiary volunteers. *Human Computation*, 1(2), December 2014.

[240] Charlene Jennett, Laure Kloetzer, Daniel Schneider, Ioanna Iacovides, Anna Cox, Margaret Gold, Brian Fuchs, Alexandra Eveleigh, Kathleen Mathieu, Zoya Ajani, and Yasmin Talsi. Motivations, learning and creativity in online citizen science. *Journal of Science Communication*, 15(03):A05, April 2016.

[241] Charlene Jennett, Laure Kloetzer, Daniel Schneider, Ioanna Iacovides, Anna Cox, Margaret Gold, Brian Fuchs, Alexandra Eveleigh, Kathleen Methieu, Zoya Ajani, et al. Motivations, learning and creativity in online citizen science. *Journal of Science Communication*, 15(3), 2016.

[242] Jesper Hasseriis Mohr Jensen, Miroslav Gajdacz, Shaeema Zaman Ahmed, Jakub Herman Czarkowski, Carrie Weidner, Janet Rafner, Jens Jakob Sørensen, Klaus Mølmer, and Jacob Friis Sherson. Crowdsourcing human common sense for quantum control. *Physical Review Research*, 3(1):013057, January 2021.

[243] Keren Jiang, Di Zhang, Tsubasa Iino, Risa Kimura, Tatsuo Nakajima, Kana Shimizu, Masahito Ohue, and Yutaka Akiyama. A playful tool for predicting protein-protein docking. In *Proceedings of the 18th International Conference on Mobile and Ubiquitous Multimedia*, pages 1–5, 2019.

[244] Ting Jiang. Cheating in mind games: The subtlety of rules matters. *Journal of Economic Behavior & Organization*, 93:328–336, 2013.

[245] Julien Thiennot. *Cookie Clicker*. Game [Android, Windows, Web browser], 2013. DashNet, Playsaurus.

[246] Jae-Eun Jung and Yang-Hee Nam. Types of gameplay mechanics in human computation games. *Journal of Korea Game Society*, 15(6):157–170, 2015.

[247] Jesper Juul. *Half-real: Video games between real rules and fictional worlds*. MIT press, 2011.

[248] Daniel Kahneman. *Thinking, fast and slow*. Macmillan, 2011.

[249] Slava Kalyuga and Jan L Plass. Evaluating and managing cognitive load in games. In *Handbook of research on effective electronic gaming in education*, pages 719–737. IGI Global, 2009.

[250] Sean HK Kang. Spaced repetition promotes efficient and effective learning: Policy implications for instruction. *Policy Insights from the Behavioral and Brain Sciences*, 3(1):12–19, 2016.

[251] Pavan Kantharaju, Katelyn Alderfer, Jichen Zhu, Bruce Char, Brian Smith, and Santiago Ontanón. Tracing player knowledge in a parallel programming educational game. In *Fourteenth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2018.

[252] Karl M Kapp. *The gamification of learning and instruction: game-based methods and strategies for training and education*. John Wiley & Sons, 2012.

[253] Leo Juri Kaufmann. *Organizational Coordination of Digital Structures: The Effects of ICT and Values on Grand Challenges*. PhD thesis, Technische Universität Kaiserslautern, 2022.

[254] Alexander Kawrykow, Gary Roumanis, Alfred Kam, Daniel Kwak, Clarence Leung, Chu Wu, Eleyine Zarour, Luis Sarmenta, Mathieu Blanchette, Jérôme Waldispühl, et al. Phylo: a citizen science approach for improving multiple sequence alignment. *PloS one*, 7(3):e31362, 2012.

[255] Benjamin Ernest Keep. *Becoming Expert Problem Solvers: A Case Study in what Develops and how*. PhD thesis, Stanford University, 2018.

[256] John M Keller. Development and use of the arcs model of instructional design. *Journal of instructional development*, 10(3):2, 1987.

[257] Ashley Rose Kelly and Kate Maddalena. Harnessing agency for efficacy:"foldit" and citizen science. *Poroi*, 11(1):1–20, 2015.

[258] Joke Kenens, Michiel Van Oudheusden, Go Yoshizawa, and Ine Van Hoyweghen. Science by, with and for citizens: rethinking 'citizen science'after the 2011 fukushima disaster. *Palgrave communications*, 6(1):1–8, 2020.

[259] Itay Keren. Teaching by design: Tips for effective tutorials from "mushroom 11.", 2017.

[260] Firas Khatib, Seth Cooper, Michael D Tyka, Kefan Xu, Ilya Makedon, Zoran Popović, and David Baker. Algorithm discovery by protein folding game players. *Proceedings of the National Academy of Sciences*, 108(47):18949–18953, 2011.

[261] Firas Khatib, Frank DiMaio, Foldit Contenders Group, Foldit Void Crushers Group, Seth Cooper, Maciej Kazmierczyk, Miroslaw Gilski, Szymon Krzywda, Helena Zabranska, Iva Pichova, et al. Crystal structure of a monomeric retroviral protease solved by protein folding game players. *Nature structural & molecular biology*, 18(10):1175–1177, 2011.

[262] Louise H Kidder and Michelle Fine. Qualitative and quantitative methods: When stories converge. *New directions for program evaluation*, 1987(35):57–75, 1987.

[263] Kristian Kiili. Digital game-based learning: Towards an experiential gaming model. *The Internet and higher education*, 8(1):13–24, 2005.

[264] Jeffrey Kim, Elan Lee, Timothy Thomas, and Caroline Dombrowski. Storytelling in new media: The case of alternate reality games, 2001–2009. *First Monday*, 2009.

[265] Jinseop S Kim, Matthew J Greene, Aleksandar Zlateski, Kisuk Lee, Mark Richardson, Srinivas C Turaga, Michael Purcaro, Matthew Balkam, Amy Robinson, Bardia F Behabadi, et al. Space–time wiring specificity supports direction selectivity in the retina. *Nature*, 509(7500):331–336, May 2014.

[266] David Kirschner and J Patrick Williams. Experts and novices or expertise? positioning players through gameplay reviews. In *DiGRA Conference*, pages 1–18, Finland, 2013. DiGRA.

[267] David Kirschner and J Patrick Williams. Measuring video game engagement through gameplay reviews. *Simulation & Gaming*, 45(4-5):593–610, 2014.

[268] Robert Kleffner, Jeff Flatten, Andrew Leaver-Fay, David Baker, Justin B Siegel, Firas Khatib, and Seth Cooper. Foldit standalone: a video game-derived protein structure manipulation interface using rosetta. *Bioinformatics*, 33(17):2765–2767, 2017.

[269] Laure Kloetzer, Julia Lorke, Joseph Roche, Yaela Golumbic, Silvia Winter, and Aiki Jõgeva. Learning in citizen science. *The science of citizen science*, 283, 2021.

[270] Eric Klopfer, Jason Haas, Scot Osterweil, and Louisa Rosenheck. *Resonant games: Design principles for learning games that connect hearts, minds, and the everyday*. MIT Press, Cambridge, MA, USA, 2018.

[271] Eric Klopfer and Scot Osterweil. The Boom and Bust and Boom of Educational Games. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Zhigeng Pan, Adrian David Cheok, Wolfgang Müller, and Fotis Liarokapis, editors, *Transactions on Edutainment IX*, volume 7544, pages 290–296. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. Series Title: Lecture Notes in Computer Science.

[272] Kenneth R Koedinger, Julie L Booth, and David Klahr. Instructional complexity and the science to constrain it. *Science*, 342(6161):935–937, 2013.

[273] Matthew Koehler and Punya Mishra. What is technological pedagogical content knowledge (tpack)? *Contemporary issues in technology and teacher education*, 9(1):60–70, 2009.

[274] Brian Koepnick. Coronavirus binder designs queued for testing! *Foldit* Blog post, 2020. https:// fold.it/portal/node/2009216.

[275] Hannu Korhonen and Elina M. I. Koivisto. Playability heuristics for mobile games. In *Proceedings of the 8th Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '06, pages 9–16, New York, NY, USA, 2006. ACM.

[276] Hannu Korhonen and Elina MI Koivisto. Playability heuristics for mobile multi-player games. In *Proceedings of the 2nd international conference on Digital interactive media in entertainment and arts*, pages 28–35, New York, NY, USA, 2007. ACM, ACM.

[277] Klaus Krippendorff. Computing krippendorff's alpha-reliability. Working paper, University of Pennsylvania, 2011. Retrieved from https://repository.upenn.edu/asc_papers/43/.

[278] KST Studio. *Sea Bubble*. Game [Windows], 2019. Beltane Games.

[279] Jeff Kuhn. Minecraft: Education edition. *calico journal*, 35(2):214–223, 2017.

[280] Liting Kway and Alex Mitchell. Emotional agency in storygames. In *Proceedings of the 13th International Conference on the Foundations of Digital Games*, pages 1–10, New York, NY, USA, 2018. ACM.

[281] Mathieu Lafourcade, Alain Joubert, and Nathalie Le Brun. Unclassifiable gwaps. *Games with a Purpose (Gwaps)*, pages 73–89, 2015.

[282] Anne Land-Zandstra, Gaia Agnello, and Yaşar Selman Gültekin. Participants in citizen science. *The Science of Citizen Science*, page 243, 2021.

[283] Anne M. Land-Zandstra, Mara M. van Beusekom, Carl E. Koppeschaar, and Jos M. van den Broek. Motivation and learning impact of Dutch flu-trackers. *JCOM*, 15(01 A04):1–26, 2016.

[284] Diana Laurillard. *Teaching as a design science: Building pedagogical patterns for learning and technology*. Routledge, 2013.

[285] Nilli Lavie, Aleksandra Hirst, Jan W De Fockert, and Essi Viding. Load theory of selective attention and cognitive control. *Journal of experimental psychology: General*, 133(3):339, 2004.

[286] Linda Costigan Lederman. Debriefing: Toward a Systematic Assessment of Theory and Practice. *Simulation & Gaming*, 23(2):145–160, June 1992.

[287] Jeehyung Lee, Wipapat Kladwang, Minjae Lee, Daniel Cantu, Martin Azizyan, Hanjoo Kim, Alex Limpaecher, Snehal Gaikwad, Sungroh Yoon, Adrien Treuille, et al. Rna design rules from a massive open laboratory. *Proceedings of the National Academy of Sciences*, 111(6):2122–2127, February 2014.

[288] Kathryn A. Lee, Jonathan R. Lee, and Patrick Bell. A review of Citizen Science within the Earth Sciences: potential benefits and obstacles. *Proceedings of the Geologists' Association*, 131(6):605–617, December 2020.

[289] Kisuk Lee. Crowd intelligence in eyewire. Technical report, 2015. Retrieved from `http://kisuklee.wikidot.com/research#toc2` on May 6, 2021.

[290] Lukas Legner, Chloe Eghtebas, and Gudrun Klinker. Persuasive mobile game mechanics for user retention. In *Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, pages 493–500, 2019.

[291] Hjalti Leifsson and Jóhann Örn Bjarkason. *Project Discovery-Advancing scientific research by implementing citizen science in EVE Online*. PhD thesis, University of Reykjavík, 2015.

[292] Susan Leigh Star. This is not a boundary object: Reflections on the origin of a concept. *Science, Technology, & Human Values*, 35(5):601–617, 2010.

[293] Jimmie Leppink, Fred Paas, Tamara Van Gog, Cees PM van Der Vleuten, and Jeroen JG Van Merrienboer. Effects of pairs of problems and examples on task performance and different types of cognitive load. *Learning and Instruction*, 30:32–42, 2014.

[294] Jimmie Leppink and Angelique van den Heuvel. The evolution of cognitive load theory and its application to medical education. *Perspectives on medical education*, 4(3):119–127, 2015.

[295] Calvin A Liang, Sean A Munson, and Julie A Kientz. Embracing four tensions in human-computer interaction research with marginalized people. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 28(2):1–47, 2021.

[296] Andreas Lieberoth, Mads Kock Pedersen, Andreea Catalina Marin, Tilo Planke, and Jacob Friis Sherson. Getting humans to do quantum optimization-user acquisition, engagement and early results from the citizen cyberscience game quantum moves. *arXiv preprint arXiv:1506.08761*, 2015.

[297] Liftoff. Mobile app trends report. https://liftoff.io/wp-content/uploads/2019/10/2019-Mobile-App-Trends-Report.pdf, 2019.

[298] Jung Lim, Robert A Reiser, and Zane Olina. The effects of part-task and whole-task instructional approaches on acquisition and transfer of a complex cognitive skill. *Educational Technology Research and Development*, 57(1):61–77, 2009.

[299] Chun-Yi Lin and Charles M Reigeluth. Scaffolding learner autonomy in a wiki-supported knowledge building community and its implications for mindset change. *British Journal of Educational Technology*, 50(5):2667–2684, 2019.

[300] John Lindstedt and Wayne Gray. Extreme expertise: Exploring expert behavior in tetris. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Berlin, Germany, 2013. John Wiley & Sons.

[301] Shlomit Aharoni Lir. Strangers in a seemingly open-to-all website: the gender bias in wikipedia. *Equality, Diversity and Inclusion: An International Journal*, 2019.

[302] Derek Lomas, Kishan Patel, Jodi L Forlizzi, and Kenneth R Koedinger. Optimizing challenge in an educational game using large-scale design experiments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 89–98, 2013.

[303] J Derek Lomas, Kenneth Koedinger, Nirmal Patel, Sharan Shodhan, Nikhil Poonwala, and Jodi L Forlizzi. Is difficulty overrated?: The effects of choice, novelty and suspense on intrinsic motivation in educational games. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 1028–1039. ACM, 2017.

[304] Yanjin Long and Vincent Aleven. Educational game and intelligent tutoring system: A classroom study and comparative design analysis. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 24(3):1–27, 2017.

[305] Jacqueline Low. A pragmatic definition of the concept of theoretical saturation. *Sociological Focus*, 52(2):131–139, 2019.

[306] Luden.io. *InMind VR*. Game [Windows, macOS, Linux, Android, iOS], 2015. Luden.io.

[307] Luden.io. *while True: learn()*. Game [Windows, macOS, Linux], 2019. Luden.io.

[308] Miguel Angel Luengo-Oroz, Asier Arranz, and John Frean. Crowdsourcing malaria parasite quantification: an online game for analyzing images of infected thick blood smears. *Journal of medical Internet research*, 14(6):e167, 2012.

[309] Oisin Mac Aodha, Rory Gibb, Kate E Barlow, Ella Browning, Michael Firman, Robin Freeman, Briana Harder, Libby Kinsey, Gary R Mead, Stuart E Newson, et al. Bat detective—deep learning tools for bat acoustic signal detection. *PLoS computational biology*, 14(3):e1005995, 2018.

[310] Joseph Macey. Heuristics for evaluating video games: A two-tier set incorporating universal and genre-specific elements, 2016. Master's thesis. Tampere University.

[311] Colleen Macklin and John Sharp. *Games, design and play: a detailed approach to iterative game design*. Addison-Wesley, Boston, MA ; San Francisco, CA, first edition edition, 2016.

[312] Kathleen M MacQueen, Eleanor McLellan, Kelly Kay, and Bobby Milstein. Codebook development for team-based qualitative analysis. *Cam Journal*, 10(2):31–36, 1998.

[313] Thomas W Malone. Heuristics for designing enjoyable user interfaces: Lessons from computer games. In *Proceedings of the 1982 conference on Human factors in computing systems*, pages 63–68, New York, NY, USA, 1982. ACM, ACM.

[314] Tim Marsh. Slow serious games, interactions and play: Designing for positive and serious experience and reflection. *Entertainment computing*, 14:45–53, 2016.

[315] Vivien Marx. Neuroscience waves to the crowd. *Nature methods*, 10(11):1069–1074, 2013.

[316] Richard E Mayer. Cognitive theory of multimedia learning. *The Cambridge handbook of multimedia learning*, 3148, 2005.

[317] Richard E Mayer and Roxana Moreno. A cognitive theory of multimedia learning: Implications for design principles. *Journal of Educational Psychology*, 91(2):358–368, 1998.

[318] Richard E Mayer and Roxana Moreno. Nine ways to reduce cognitive load in multimedia learning. *Educational psychologist*, 38(1):43–52, 2003.

[319] Suvodeep Mazumdar, Stuart Wrigley, and Fabio Ciravegna. Citizen science and crowdsourcing for earth observations: An analysis of stakeholder opinions on the present and future. *Remote Sensing*, 9(1):87, 2017.

[320] Benedict McAteer, Wesley Flannery, and Brendan Murtagh. Linking the motivations and outcomes of volunteers to understand participation in marine community science. *Marine Policy*, 124:104375, February 2021.

[321] Edward McAuley, Terry Duncan, and Vance V Tammen. Psychometric properties of the intrinsic motivation inventory in a competitive sport setting: A confirmatory factor analysis. *Research quarterly for exercise and sport*, 60(1):48–58, 1989.

[322] McGill University. *Phylo*. Game [Web Browser, Android, iOS], 2010. `https://phylo.cs.mcgill.ca/`.

[323] McGill University. *Colony B*. Game [iOS, Android], 2018. `http://csb.cs.mcgill.ca/colonyb/`.

[324] Elisa D. Mekler, Florian Brühlmann, Alexandre N. Tuch, and Klaus Opwis. Towards understanding the effects of individual gamification elements on intrinsic motivation and performance. *Computers in Human Behavior*, 71:525–534, 2017.

[325] M David Merrill. First principles of instruction. *Educational technology research and development*, 50(3):43–59, 2002.

[326] Jacob Mikkelsen and Mette Pødenphant. Level design in 'hitman': Guiding players in a non-linear sandbox, 2016.

[327] Colin Milburn, Katherine Buse, Ranjodh Singh Dhaliwal, Melissa Wills, Raida Aldosari, Patrick Camarador, Josh Aaron Miller, and Justin Siegel. Join the fold: Video games, science fiction, and the refolding of citizen science. *Design Issues*, 39(1):70–87, 2023.

[328] George A Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

[329] Josh Aaron Miller. Roleplaying as a solution to the quarterbacking problem problem of cooperative and educational games. *Analog Game Studies*, 9(4):online, 2021.

[330] Josh Aaron Miller, Katherine Buse, Ranjodh Singh Dhaliwal, Justin Siegel, Seth Cooper, and Colin Milburn. Wrapped in story: The affordances of narrative for citizen science games. In *Proceedings of the Foundations of Digital Games*, page In press, New York, NY, USA, 2023. ACM.

[331] Josh Aaron Miller and Seth Cooper. Case studies in game-based complex learning. *Multimodal Technologies and Interaction*, 5(12):72, 2021.

[332] Josh Aaron Miller and Seth Cooper. Barriers to expertise in citizen science games. In *CHI Conference on Human Factors in Computing Systems*, pages 1–25, New York, NY, USA, 2022. ACM.

[333] Josh Aaron Miller, Kutub Gandhi, Anna Gander, and Seth Cooper. A survey of citizen science gaming experiences. *Citizen Science: Theory and Practice*, 7(1):1–12, 2022.

[334] Josh Aaron Miller, Britton Horn, Matthew Guthrie, Jonathan Romano, Guy Geva, Celia David, Amy Robinson Sterling, and Seth Cooper. How do players and developers of citizen science games conceptualize skill chains? In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, CHI PLAY '21, page in press, New York, NY, USA, 2021. Association for Computing Machinery.

[335] Josh Aaron Miller, Firas Khatib, Haley Hammond, Seth Cooper, and Scott Horowitz. Introducing foldit education mode. *Nature structural & molecular biology*, 27(9):769–770, 2020.

[336] Josh Aaron Miller, Vivian Lee, Seth Cooper, and Magy Seif El-Nasr. Large-scale analysis of visualization options in a citizen science game. In *Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, pages 535–542, 2019.

[337] Josh Aaron Miller, Uttkarsh Narayan, Matthew Hantsbarger, Seth Cooper, and Magy Seif El-Nasr. Expertise and engagement: re-designing citizen science games with players' minds in mind. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, pages 1–11, New York, NY, USA, 2019. ACM.

[338] Josh Aaron Miller, Libuše Hannah Vepřek, Sebastian Deterding, and Seth Cooper. Practical recommendations from a multi-perspective needs and challenges assessment of citizen science games. *PLOS ONE*, page In press, 2023.

[339] Abraham J Miller-Rushing. Reflections on the strong growth of citizen science: An interview with abe miller-rushing. *Maine Policy Review*, 26(2):92–93, 2017.

[340] Moffitt Cancer Center. *Cancer Crusade*. Game [iOS, Android], 2018. `https://cancercrusadegame.com/`.

[341] Mojang Studios. *Minecraft*, 2009.

[342] Christian Mönch, Gisle Grimen, and Roger Midtstraum. Protecting online games against cheating. In *Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games*, 2006.

[343] Aidan Moran, Mark Campbell, and John Toner. Exploring the cognitive mechanisms of expertise in sport: Progress and prospects. *Psychology of Sport and Exercise*, 42:8–15, 2019.

[344] Raphaël Morin, Pierre-Majorique Léger, Sylvain Senecal, Marie-Christine Bastarache-Roberge, Mario Lefèbrve, and Marc Fredette. The effect of game tutorial: A comparison between casual and hardcore gamers. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, CHI PLAY Companion '16, page 229–237, New York, NY, USA, 2016. Association for Computing Machinery.

[345] Briana B Morrison and Betsy DiSalvo. Khan academy gamifies computer science. In *Proceedings of the 45th ACM technical symposium on Computer science education*, pages 39–44, 2014.

[346] Benedikt Morschheuser, Juho Hamari, and Jonna Koivisto. Gamification in crowdsourcing: a review. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pages 4375–4384. IEEE, 2016.

[347] Benedikt Morschheuser, Juho Hamari, Jonna Koivisto, and Alexander Maedche. Gamified crowdsourcing: Conceptualization, literature review, and future agenda. *International Journal of Human-Computer Studies*, 106:26–43, 2017.

[348] MumboJumbo. *Luxor Evolved*. Game [Windows], 2012. MumboJumbo.

[349] Andrew Murphy. A gamer's guide to eterna, 2020. Retrieved from `https://www.keystone-research-solutions.com/en/item/a-gamers-guide-to-eterna` on November 29, 2021.

[350] K Denise Muth. Effects of cuing on middle-school students' performance on arithmetic word problems containing extraneous information. *Journal of Educational Psychology*, 83(1):173, 1991.

[351] LE Nacke, Anders Drachen, and Stefan Göbel. Methods for evaluating gameplay experience in a serious gaming context. *International Journal of Computer Science in Sport*, 9(2):1–12, 2010.

[352] Lennart E Nacke and Christoph Sebastian Deterding. The maturing of gamification research. *Computers in Human Behaviour*, pages 450–454, 2017.

[353] Lawrence J Najjar. Principles of educational multimedia user interface design. *Human factors*, 40(2):311–323, 1998.

[354] Namco Limited. *Pac-Man*. Game [Arcade], July 1980. Namco Limited, Midway Games Inc.

[355] NCCR QSIT, University of Basel. *Decodoku*. Game [Web Browser, iOS, Android], 2016. `http://www.decodoku.com/`.

[356] Ndemic Creations. *Plague Inc: Evolved*. Game [Windows, macOS, Linux, Nintendo Switch], 2016. Ndemic Creations.

[357] Neural Reckoning and Pawar Lab. *EcoBuilder*. Game [Android, iOS], 2020. `https://ecobuildergame.org/`.

[358] Greg Newman, Andrea Wiggins, Alycia Crall, Eric Graham, Sarah Newman, and Kevin Crowston. The future of citizen science: emerging technologies and shifting paradigms. *Frontiers in Ecology and the Environment*, 10(6):298–304, August 2012.

[359] NEXT Studios. *Unheard — Voices of Crime*. Game [Windows, macOS], 2019. NEXT Studios, bilibili.

[360] Jakob Nielsen. Enhancing the explanatory power of usability heuristics. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 152–158, New York, NY, USA, 1994. ACM, ACM.

[361] Jörg Niewöhner. Co-laborative anthropology: crafting reflexivities experimentally. *Analysis and Interpretation. Helsinki: Ethnos*, pages 81–125, 2016.

[362] Nintendo EPD. *The Legend of Zelda: Breath of the Wild*. Game [Nintendo Switch, Wii U], March 2017. Nintendo Co. Ltd., Kyoto, Japan.

[363] Nintendo R&D4. *Super Mario Bros.* Game [Nintendo Entertainment System, Arcade], September 1985. Nintendo.

[364] Donald A Norman and Stephen W Draper. *User centered system design: New perspectives on human-computer interaction*. CUMINCAD, 1986.

[365] Jill Nugent. inaturalist: citizen science for 21st-century naturalists. *Science Scope*, 41(7):12, 2018.

[366] Thomas E Nygren. Psychometric properties of subjective workload measurement techniques: Implications for their use in the assessment of perceived mental workload. *Human Factors*, 33(1):17–33, 1991.

[367] Martin Oliver and Diane Carr. Learning in virtual worlds: Using communities of practice to explain how people learn from play. *British Journal of Educational Technology*, 40(3):444–457, 2009.

[368] Open AI. Gpt-4 technical report. Technical report, Open AI, 2023. https://cdn.openai.com/papers/gpt-4.pdf.

[369] Rita Orji, Regan L Mandryk, and Julita Vassileva. Improving the efficacy of games for change using personalization models. *ACM Trans. Comput.-Hum. Interact.*, 24(5):32:1–32:22, October 2017.

[370] Rita Orji, Regan L Mandryk, Julita Vassileva, and Kathrin M Gerling. Tailoring persuasive health games to gamer type. In *Proceedings of the sigchi conference on human factors in computing systems*, pages 2467–2476, 2013.

[371] Rita Orji, Lennart E Nacke, and Chrysanne Di Marco. Towards personality-driven persuasive health games and gamified systems. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 1015–1027, 2017.

[372] David Osborne. Tales from a crowdsourced game design, 2016.

[373] Oxymoron Games. *Project Hospital*. Game [Windows, macOS, Linux], 2018. Oxymoron Games.

[374] Janne Paavilainen. Critical review on video game evaluation heuristics: Social games perspective. In *Proceedings of the International Academic Conference on the Future of Game Design and Technology*, Futureplay '10, pages 56–65, New York, NY, USA, 2010. ACM.

[375] Janne Paavilainen. Defining playability of games: functionality, usability, and gameplay. In *Proceedings of the 23rd International Conference on Academic Mindtrek*, pages 55–64, 2020.

[376] Carole Paleco, Sabina García Peter, Nora Salas Seoane, Julia Kaufmann, and Panagiota Argyri. Inclusiveness and diversity in citizen science. *The Science of Citizen Science*, page 261, 2021.

[377] Kunwoo Park, Meeyoung Cha, Haewoon Kwak, and Kuan-Ta Chen. Achievement and friends: key factors of player retention vary across player levels in online multiplayer games. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 445–453. International World Wide Web Conferences Steering Committee, 2017.

[378] Julia K Parrish, Timothy Jones, Hillary K Burgess, Yurong He, Lucy Fortson, and Darlene Cavalier. Hoping for optimality or designing for inclusion: Persistence, learning, and the social network of citizen science. *Proceedings of the National Academy of Sciences*, 116(6):1894–1901, 2019.

[379] Mario Passalacqua, Raphaël Morin, Sylvain Sénécal, Lennart E Nacke, and Pierre-Majorique Léger. Demystifying the first-time experience of mobile games: The presence of a tutorial has a positive impact on non-expert players' flow and continuous-use intentions. *Multimodal Technologies and Interaction*, 4(3):41, 2020.

[380] Katherine Payne, Mark J Keith, Ryan M Schuetzler, and Justin Scott Giboney. Examining the learning effects of live streaming video game instruction over twitch. *Computers in Human Behavior*, 77:95–109, 2017.

[381] Ei Pa Pa Pe-Than, Dion Hoe-Lian Goh, and Chei Sian Lee. A typology of human computation games: an analysis and a review of current games. *Behaviour & Information Technology*, 34(8):809–824, 2015.

[382] Mads Kock Pedersen, Carlos Mauricio Castaño Díaz, Mario Alejandro Alba-Marrugo, Ali Amidi, Rajiv Vaid Basaiawmoit, Carsten Bergenholtz, Morten H Christiansen, Miroslav Gajdacz, Ralph Hertwig, Byurakn Ishkhanyan, et al. Cognitive abilities in the wild: Population-scale game-based cognitive assessment. *arXiv preprint arXiv:2009.05274*, April 2020. arXiv: 2009.05274.

[383] Maite Pelacho, Hannot Rodríguez, Fernando Broncano, Renata Kubus, Francisco Sanz García, Beatriz Gavete, and Antonio Lafuente. Science as a Commons: Improving the Governance of Knowledge Through Citizen Science. *The Science of Citizen Science*, page 57, 2021. Publisher: Springer Nature.

[384] Wei Peng, Jih-Hsuan Lin, Karin A Pfeiffer, and Brian Winn. Need satisfaction supportive game features as motivational determinants: An experimental study of a self-determination theory guided exergame. *Media Psychology*, 15(2):175–196, 2012.

[385] Andrew Pickering. *The mangle of practice: Time, agency, and science*. University of Chicago Press, 2010.

[386] David Pinelle, Nelson Wong, Tadeusz Stach, and Carl Gutwin. Usability heuristics for networked multiplayer games. In *Proceedings of the ACM 2009 international conference on Supporting group work*, pages 169–178, New York, NY, USA, 2009. ACM, ACM.

[387] Pixel Maniacs. *ChromaGun*. Game [Windows, macOS, Linux], 2016. Pixel Maniacs.

[388] Jan L Plass, Bruce D Homer, and Charles K Kinzer. Foundations of game-based learning. *Educational psychologist*, 50(4):258–283, 2015.

[389] Polytechnic University of Madrid. *MalariaSpot*. Game [Web Browser, iOS, Android], 2015. `https://malariaspot.org/en/`.

[390] Polytechnic University of Madrid, Spotlab. *MalarisSpot Bubbles*. Game [Web Browser, iOS, Android], 2018. `https://bubbles.malariaspot.org/`.

[391] Lesandro Ponciano and Francisco Brasileiro. Finding volunteers' engagement profiles in human computation for citizen science projects. *arXiv preprint arXiv:1501.02134*, 2015.

[392] Marisa Ponti, Thomas Hillman, Christopher Kullenberg, and Dick Kasperowski. Getting it right or being top rank: Games in citizen science. *Citizen Science: Theory and Practice*, 3(1), 2018.

[393] Marisa Ponti, Thomas Hillman, and Igor Stankovic. Science and gamification: The odd couple? In *Proceedings of the 2015 annual symposium on computer-human interaction in play*, pages 679–684, 2015.

[394] Marisa Ponti and Igor Stankovic. Scripts in a Gamified Crowdsourcing Approach to Do Science. In *Proceedings of the European Conference on Computer-Supported Co-operative Work -– ECSCW*, Oslo, Norway, September 2015. Springer. Retrieved from https://marisaponti.files.wordpress.com/2015/03/ecscw_ws-mp_final.pdf.

[395] Marisa Ponti, Igor Stankovic, Wolmet Barendregt, Bruno Kestemont, and Lyn Bain. Chefs know more than just recipes: Professional vision in a citizen science game. *Human Computation*, 5:1–12, 2018.

[396] Lev Poretski and Anthony Tang. Press a to jump: Design strategies for video game learnability. In *CHI Conference on Human Factors in Computing Systems*, pages 1–26, New York, NY, USA, 2022. ACM.

[397] Fernando Portela. An unexpectedly effective monte carlo technique for the rna inverse folding problem. *BioRxiv*, page 345587, 2018.

[398] James Portnow. Tutorials 101 - how to design a good game tutorial, April 2012.

[399] Marc Prensky. Digital game-based learning. *Computers in Entertainment*, 1(1):21–21, October 2003.

[400] Nathan Prestopnik and Kevin Crowston. Purposeful gaming & socio-computational systems: a citizen science design case. In *Proceedings of the 17th ACM international conference on Supporting group work*, pages 75–84, 2012.

[401] Nathan Prestopnik, Kevin Crowston, and Jun Wang. Exploring data quality in games with a purpose. *iConference 2014 Proceedings*, pages 213–228, 2014.

[402] Nathan Prestopnik, Kevin Crowston, and Jun Wang. Gamers, citizen scientists, and data: Exploring participant contributions in two games with a purpose. *Computers in Human Behavior*, 68:254–268, 2017.

[403] Nathan Prestopnik and Dania Souid. Forgotten island: a story-driven citizen science adventure. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, pages 2643–2646. ACM Press, Paris, France, 2013.

[404] Nathan R Prestopnik. Theory, design and evaluation–(don't just) pick any two. *AIS Transactions on Human-Computer Interaction*, 2(4):167–177, 2010.

[405] Nathan R Prestopnik and Kevin Crowston. Gaming for (citizen) science: Exploring motivation and data quality in the context of crowdsourced science through the design and evaluation of a social-computational system. In *2011 IEEE Seventh International Conference on e-Science Workshops*, pages 28–33. IEEE, 2011.

[406] Nathan R Prestopnik and Kevin Crowston. Citizen science system assemblages: understanding the technologies that support crowdsourced science. In *Proceedings of the 2012 iConference*, pages 168–176. iSchools, Inc., 2012.

[407] Nathan R Prestopnik and Jian Tang. Points, stories, worlds, and diegesis: Comparing player experiences in two citizen science games. *Computers in Human Behavior*, 52:492–506, 2015.

[408] Princeton University. *Eyewire*. Game [Web Browser], 2012. `https://eyewire.org/explore`.

[409] Andrew K Przybylski, C Scott Rigby, and Richard M Ryan. A motivational model of video game engagement. *Review of general psychology*, 14(2):154, 2010.

[410] Victor Punegov. The art of fair play: Developing the best systems to deal with players who cheat, October 2021.

[411] Yu Qi Qiao, Jun Shen, Xiao Liang, Song Ding, Fang Yuan Chen, Li Shao, Qing Zheng, and Zhi Hua Ran. Using cognitive theory to facilitate medical education. *BMC medical education*, 14(1):79, 2014.

[412] Quiet River. *Zup! 2*. Game [Windows], 2016. Quiet River.

[413] Chris Quintana, Brian J Reiser, Elizabeth A Davis, Joseph Krajcik, Eric Fretz, Ravit Golan Duncan, Eleni Kyza, Daniel Edelson, and Elliot Soloway. A scaffolding design framework for software to support science inquiry. *The journal of the learning sciences*, 13(3):337–386, 2004.

[414] M Jordan Raddick, Georgia Bracey, Pamela L Gay, Chris J Lintott, Carie Cardamone, Phil Murray, Kevin Schawinski, Alexander S Szalay, and Jan Vandenberg. Galaxy zoo: Motivations of citizen scientists. *arXiv preprint arXiv:1303.6886*, 2013.

[415] M Jordan Raddick, Georgia Bracey, Pamela L Gay, Chris J Lintott, Phil Murray, Kevin Schawinski, Alexander S Szalay, and Jan Vandenberg. Galaxy zoo: Exploring the motivations of citizen science volunteers. *arXiv preprint arXiv:0909.2925*, 2009.

[416] Janet Rafner, Miroslav Gajdacz, Gitte Kragh, Arthur Hjorth, Anna Gander, Blanka Palfi, Aleks Berditchevskaia, François Grey, Kobi Gal, Avi Segal, et al. Revisiting citizen science through the lens of hybrid intelligence. *arXiv preprint arXiv:2104.14961*, 2021.

[417] Tzipora Rakedzon, Elad Segev, Noam Chapnik, Roy Yosef, and Ayelet Baram-Tsabari. Automatic jargon identifier for scientists engaging with the public and science communication educators. *PloS one*, 12(8):e0181742, 2017.

[418] Rami Ismail (رامي) [@tha_rami]. Based on the feedback, I'm finally done structuring that thing I asked about last month where indie devs can book small advice calls. I ended up with a free 15-minute call, a €15 30-minute call, a €100 60 minute call. There's additional free options for marginalized developers., June 2020.

[419] Andrés Ramirez Sessarego and Felype Arévalo Arancibia. Making sense of a game: A look into tutorials and character mechanics, 2019.

[420] Ruziana Binti Mohamad Rasli, Faudziah Ahmad, and Siti Sakira Kamaruddin. A comparative study of conceptual graph and concept map. *Journal of Engineering and Applied Sciences*, 9(9):1442–1446, 2014.

[421] Lisa M. Rasmussen and Caren Cooper. Citizen Science Ethics. *Citizen Science: Theory and Practice*, 4(1):5, March 2019.

[422] Sheri Graner Ray. Tutorials: Learning to play, 2010.

[423] Razbuten. What games are like for someone who doesn't play games, 2019.

[424] Re-Logic. *Terraria*. Game [Windows, PlayStation 3, Xbox 360, PlayStation Vita, iOS, Android, Windows Phone, PlayStation 4, PlayStation 5, Xbox One, Xbox Series X/S, macOS, Linux, Nintendo 3DS, Wii U, Nintendo Switch, Google Stadia], May 2011. 505 Games.

[425] Neal Reeves, Peter West, and Elena Simperl. A game without competition is hardly a game": The impact of competitions on player activity in a human computation game. In *Sixth AAAI Conference on Human Computation and Crowdsourcing*, 2018.

[426] C Reigeluth and R Stein. Elaboration theory. *Instructional-design theories and models: An overview of their current status (1983)*, pages 335–381, 1983.

[427] David B. Resnik. Citizen Scientists as Human Subjects: Ethical Issues. *Citizen Science: Theory and Practice*, 4(1):11, March 2019.

[428] David B. Resnik, Kevin C. Elliott, and Aubrey K. Miller. A framework for addressing ethical issues in citizen science. *Environmental Science & Policy*, 54:475–481, December 2015.

[429] Günter Daniel Rey and Florian Buchwald. The expertise reversal effect: Cognitive load and motivational explanations. *Journal of Experimental Psychology: Applied*, 17(1):33, 2011.

[430] Hauke Riesch and Clive Potter. Citizen science as seen by scientists: Methodological, epistemological and ethical dimensions. *Public Understanding of Science*, 23(1):107–120, January 2014.

[431] Scott Rigby. The freedom fallacy: Understanding ”player autonomy” in game design, 2017.

[432] Scott Rigby and Richard Ryan. The player experience of need satisfaction (pens) model. *Immersyve Inc*, pages 1–22, 2007.

[433] Rob Howland. *Influent*. Game [Windows, macOS, Linux], March 2014. Three Flip Studios.

[434] Lucy Danielle Robinson, Jade Lauren Cawthray, Sarah Elizabeth West, Aletta Bonn, and Janice Ansine. Ten principles of citizen science. In *Citizen science: Innovation in open science, society and policy*, pages 27–40. UCL Press, 2018.

[435] Jane Roskams and Zoran Popović. Power to the people: addressing big data challenges in neuroscience by creating a new cadre of citizen neuroscientists. *Neuron*, 92(3):658–664, 2016.

[436] Mark A. Rothstein, John T. Wilbanks, and Kyle B. Brothers. Citizen Science on Your Smartphone: An ELSI Research Agenda: Currents in Contemporary Bioethics. *Journal of Law, Medicine & Ethics*, 43(4):897–903, 2015.

[437] Dana Rotman, Jenny Preece, Jen Hammock, Kezee Procita, Derek Hansen, Cynthia Parr, Darcy Lewis, and David Jacobs. Dynamic changes in motivation in collaborative citizen-science projects. In *Proceedings of the ACM 2012 conference on computer supported cooperative work*, pages 217–226, 2012.

[438] Thomas A Rowles. Power to the people: Does eterna signal the arrival of a new wave of crowd-sourced projects?, 2013.

[439] Simone Rüfenacht, Tim Woods, Gaia Agnello, Margaret Gold, Philipp Hummer, Anne Land-Zandstra, and Andrea Sieber. Communication and dissemination in citizen science. *The Science of Citizen Science*, page 475, 2021.

[440] Fredrik Rusk, Matilda Ståhl, and Kenneth Silseth. Exploring peer mentoring and learning among experts and novices in online in-game interactions. In *Proceedings of the 14th International Conference on Game Based Learning, ed. P. Fotaris*, pages 461–468, 2020.

[441] Richard M Ryan and Edward L Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, 25(1):54–67, 2000.

[442] Richard M Ryan and Edward L Deci. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American psychologist*, 55(1):68, 2000.

[443] Richard M Ryan, C Scott Rigby, and Andrew Przybylski. The motivational pull of video games: A self-determination theory approach. *Motivation and emotion*, 30(4):344–360, 2006.

[444] Michael Sailer, Jan Ulrich Hense, Sarah Katharina Mayr, and Heinz Mandl. How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction. *Computers in Human Behavior*, 69:371–380, 2017.

[445] Katie Salen, Katie Salen Tekinbaş, and Eric Zimmerman. *Rules of play: Game design fundamentals*. MIT press, 2004.

[446] Malik Sallam. The utility of chatgpt as an example of large language models in healthcare education, research and practice: Systematic review on the future perspectives and potential limitations. *medRxiv*, pages 2023–02, 2023.

[447] Anurag Sarkar and Seth Cooper. Meet your match rating: providing skill information and choice in player-versus-level matchmaking. In *Proceedings of the 13th International Conference on the Foundations of Digital Games*, pages 1–8, 2018.

[448] Anurag Sarkar and Seth Cooper. Using rating arrays to estimate score distributions for player-versus-level matchmaking. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, pages 1–8, 2019.

[449] Anurag Sarkar, Michael Williams, Sebastian Deterding, and Seth Cooper. Engagement effects of player rating system-based matchmaking for level ordering in human computation

games. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*, pages 1–10, 2017.

[450] Henry Sauermann and Chiara Franzoni. Crowd science user contribution patterns and their implications. *Proceedings of the national academy of sciences*, 112(3):679–684, 2015.

[451] Jesse Schell. *The Art of Game Design: A book of lenses*. CRC press, 2008.

[452] Henrik Schoenau-Fog. The player engagement process-an exploration of continuation desire in digital games. In *Proceedings of the DiGRA Conference*, Finland, 2011. DiGRA.

[453] Margrit Schreier. *Qualitative content analysis in practice*. Sage publications, 2012.

[454] Karen Schrier. *Knowledge games: How playing games can solve problems, create insight, and make change*. Tech.edu. JHU Press, Baltimore, 2016.

[455] Karen Schrier. Designing Learning with Citizen Science and Games. *Emerging Learning Design Journal*, 4:9, 2017.

[456] Karen Schrier. Investigating typologies of games as research environments. In *Proceedings of International Conference on the Foundations of Digital Games*, pages 1–4, 2017.

[457] Karen Schrier. What's in a name? naming games that solve real-world problems. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*, pages 1–4, Hyannis Massachusetts, August 2017. ACM.

[458] Karen Schrier. Using games to solve real-world civic problems: Early insights and design principles. *Journal of Community Engagement & Higher Education*, 10(1):15, 2018.

[459] Karen Schrier. Designing games for moral learning and knowledge building. *Games and Culture*, 14(4):306–343, June 2019.

[460] Dale H Schunk. *Learning theories an educational perspective sixth edition*. Pearson, 2012.

[461] Daniel L Schwartz and John D Bransford. A time for telling. *Cognition and instruction*, 16(4):475–5223, 1998.

[462] ScienceAtHome. *Quantum Minds*. Game [Web Browser], 2017. https://www.scienceathome.org/games/quantum-minds/.

[463] ScienceAtHome. *Skill Lab: Science Detective*. Game [Web Browser, iOS, Android], 2018. `https://www.scienceathome.org/games/skill-lab-science-detective/`.

[464] ScienceAtHome. *Quantum Moves 2*. Game [Web Browser], 2020. `https://www.scienceathome.org/games/quantum-moves-2/`.

[465] scriptwelder. *Don't Escape: 4 Days to Survive*. Game [Windows, macOS], 2019. Armor Games Studios.

[466] Katie Seaborn and Deborah I Fels. Gamification in theory and action: A survey. *International Journal of human-computer studies*, 74:14–31, 2015.

[467] Thomas L Seamster, Richard E Redding, and George L Kaempf. A skill-based cognitive task analysis framework. *Cognitive task analysis*, pages 135–146, 2000.

[468] SeekSick6. *Breaking Good*. Game [Windows, macOS, Linux], 2017. SeekSick6.

[469] Serious Games Interactive. *President for a Day — Floodings*. Game [Windows, macOS], 2015. Serious Games Interactive.

[470] Hortense Serret, Nicolas Deguines, Yikweon Jang, Grégoire Lois, and Romain Julliard. Data quality and participant engagement in citizen science: comparing two approaches for monitoring pollinators in france and south korea. *Citizen Science: Theory and Practice*, 4(1):22, 2019.

[471] Amy Shannon, Acey Boyce, Chitra Gadwal, and Tiffany Barnes. Effective practices in game tutorial systems. In *Proceedings of the 8th International Conference on the Foundations of Digital Games*, pages 1–8, New York, NY, USA, 2013. Citeseer.

[472] Mike Sharples. Methods for evaluating mobile learning. *Researching mobile learning: Frameworks, tools and research designs*, pages 17–39, 2009.

[473] Bruce Shelley. Guidelines for developing successful games, August 2001.

[474] Jennifer L Shirk, Heidi L Ballard, Candie C Wilderman, Tina Phillips, Andrea Wiggins, Rebecca Jordan, Ellen McCallie, Matthew Minarchek, Bruce V Lewenstein, Marianne E Krasny, et al. Public participation in scientific research: a framework for deliberate design. *Ecology and society*, 17(2), 2012.

[475] Hillary C Shulman, Graham N Dixon, Olivia M Bullock, and Daniel Colón Amill. The effects of jargon on processing fluency, self-perceptions, and scientific engagement. *Journal of Language and Social Psychology*, 39(5-6):579–597, 2020.

[476] Nyet Moi Siew, Jolly Geofrey, and Bih Ni Lee. Students' algebraic thinking and attitudes towards algebra: the effects of game-based learning using dragonbox 12+ app. *The Research Journal of Mathematics and Technology*, 5(1):66–79, 2016.

[477] Artemis Skarlatidou, Alexandra Hamilton, Michalis Vitos, and Muki Haklay. What do volunteers want from citizen science technologies? a systematic literature review and best practice guidelines. *JCOM: Journal of Science Communication*, 18(1), 2019.

[478] Jeffrey Skolnick, Mu Gao, Hongyi Zhou, and Suresh Singh. Alphafold 2: why it works and its implications for understanding the relationships of protein sequence, structure, and function. *Journal of chemical information and modeling*, 61(10):4827–4831, 2021.

[479] Norman J Slamecka and Peter Graf. The generation effect: Delineation of a phenomenon. *Journal of experimental Psychology: Human learning and Memory*, 4(6):592, 1978.

[480] Sleepy Duck. *Learn Japanese to Survive! Knaji Combat*. Game [Windows, macOS], August 2018. River Crow Studio.

[481] Peter Smittenaar, Alexandra K. Walker, Shaun McGill, Christiana Kartsonaki, Rupesh J. Robinson-Vyas, Janette P. McQuillan, Sarah Christie, Leslie Harris, Jonathan Lawson, Elizabeth Henderson, Will Howat, Andrew Hanby, Gareth J. Thomas, Selina Bhattarai, Lisa Browning, and Anne E. Kiltie. Harnessing citizen science through mobile phone technology to screen for immunohistochemical biomarkers in bladder cancer. *British Journal of Cancer*, 119(2):220–229, July 2018.

[482] Sam Snodgrass, Omid Mohaddesi, Jack Hart, Guillermo Romera Rodriguez, Christoffer Holmgård, and Casper Harteveld. Like peas in pods: the player, environment, agents, system framework for the personalization of digital systems. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, pages 1–15, 2019.

[483] Bethesda Softworks. The elder scrolls, 1994–2020.

[484] Helen Spiers, Alexandra Swanson, Lucy Fortson, Brooke Simmons, Laura Trouille, Samantha Blickhan, and Chris Lintott. Everyone counts? design considerations in online citizen science. *Journal of Science Communication*, 18(1), 2019.

[485] Squad. *Kerbal Space Program*. Game [Windows, macOS, Linux, PlayStation 4, Xbox One, PlayStation 5, Xbox Series X/S], April 2015. Private Division.

[486] Kurt Squire. Designed cultures. In Constance Steinkuehler, Kurt Squire, and Sasha Barab, editors, *Games, learning, and society: Learning and meaning in the digital age*, chapter 2. Cambridge University Press, 2012.

[487] Stanford University. *Eterna*. Game [Web Browser, Android, iOS], 2010. `https://eternagame.org/`.

[488] Constance Steinkuehler. The mangle of play. *Games and culture*, 1(3):199–213, 2006.

[489] Constance Steinkuehler and Sean Duncan. Scientific habits of mind in virtual worlds. *Journal of Science Education and Technology*, 17(6):530–543, 2008.

[490] Osamuyimen Stewart, David Lubensky, and Juan M Huerta. Crowdsourcing participation inequality: a scout model for the enterprise domain. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 30–33, 2010.

[491] Alexander D Stoneman, Josh Aaron Miller, and Seth Cooper. Effects of player-level matchmaking methods in a live citizen science game. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, pages 199–206, Washington, DC, USA, 2022. AAAI Press.

[492] Stray Fawn Studio. *Niche — a genetics survival game*. Game [Windows, macOS, Linux], 2016. Stray Fawn Studio, WhisperGames.

[493] Subset Games. *FTL: Faster Than Light*. Game [Windows, macOS, Linux], 2012. Subset Games.

[494] Paul Suddaby. The many ways to show the player how it's done with in-game tutorials, 2012.

[495] Anne Sullivan and Anastasia Salter. A taxonomy of narrative-centric board and card games. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*, pages 1–10, New York, NY, USA, 2017. ACM.

[496] Devin P Sullivan, Casper F Winsnes, Lovisa Åkesson, Martin Hjelmare, Mikaela Wiking, Rutger Schutten, Linzi Campbell, Hjalti Leifsson, Scott Rhodes, Andie Nordgren, et al. Deep learning is combined with massive-scale citizen science to improve large-scale image classification. *Nature biotechnology*, 36(9):820–828, 2018.

[497] Aldric Sun and Neil Jones-Rodway. Teaching players: Tutorial and opening mission design for company of heroes., 2008.

[498] Suspicious Developments. *Gunpoint*. Game [Windows, macOS, Linux], 2013. Suspicious Developments.

[499] Penelope Sweetser and Peta Wyeth. Gameflow: a model for evaluating player enjoyment in games. *Computers in Entertainment (CIE)*, 3(3):3–3, 2005.

[500] John Sweller. Element interactivity and intrinsic, extraneous, and germane cognitive load. *Educational psychology review*, 22(2):123–138, 2010.

[501] John Sweller. Cognitive load theory. In *Psychology of learning and motivation*, volume 55, pages 37–76. Elsevier, 2011.

[502] Steve Swink. *Game feel: a game designer's guide to virtual sensation*. CRC Press, Boca Raton, Florida, USA, 2008.

[503] Alex Tamkin, Miles Brundage, Jack Clark, and Deep Ganguli. Understanding the capabilities, limitations, and societal impact of large language models. *arXiv preprint arXiv:2102.02503*, 2021.

[504] Jian Tang and Nathan R Prestopnik. Science gamers, citizen scientists, and dabblers: Characterizing player engagement in two citizen science games. *International Journal of Human–Computer Interaction*, pages 1–22, 2022.

[505] Nick Paul Taylor. Gamers crush algorithms in rna structure design challenge, 2016.

[506] TERC, Broderbund, and The Learning Company. *Zoombinis*. Game [Windows, macOS, Linux], 2015. Viva Media. Remake of *Logical Journey of the Zoombinis* (1996).

[507] Terry Cavanagh. *Super Hexagon*. Game [Windows, macOS, Linux, Android, iOS, BlackBerry 10], 2012. Terry Cavanagh.

[508] The Young Socratics. *Odyssey — The Story of Science*. Game [Windows, macOS], 2017. The Young Socratics.

[509] Carl Therrien. ”to get help, please press x” the rise of the assistance paradigm in video game design. In *DiGRA Conference*, pages 1–8, Finland, 2011. DiGRA.

[510] Thomas Bowker. *LYNE*. Game [Windows, macOS, Linux, Android, iOS], 2014. Thomas Bowker.

[511] Joseph J Thompson, Mark R Blair, Lihan Chen, and Andrew J Henrey. Video game telemetry as a critical tool in the study of complex skill learning. *PloS one*, 8(9):e75129, 2013.

[512] Line E Thomsen, Falko Weigert Petersen, Anders Drachen, and Pejman Mirza-Babaei. Identifying onboarding heuristics for free-to-play mobile games: A mixed methods approach. In *International Conference on Entertainment Computing*, pages 241–246, Cham, 2016. Springer, Springer International Publishing.

[513] Ramine Tinati, Markus Luczak-Roesch, Elena Simperl, and Wendy Hall. Because science is awesome: studying participation in a citizen science game. In *Proceedings of the 8th ACM Conference on Web Science*, pages 45–54, Hannover Germany, May 2016. ACM, ACM.

[514] Ramine Tinati, Markus Luczak-Roesch, Elena Simperl, and Wendy Hall. An investigation of player motivations in eyewire, a gamified citizen science project. *Computers in Human Behavior*, 73:527–540, 2017.

[515] Ramine Tinati, Max Van Kleek, Elena Simperl, Markus Luczak-Rösch, Robert Simpson, and Nigel Shadbolt. Designing for citizen data analysis: A cross-sectional case study of a multi-domain citizen science platform. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 4069–4078, 2015.

[516] Alexandra To, Elaine Fath, Eda Zhang, Safnah Ali, Catherine Kildunne, Anny Fan, Jessica Hammer, and Geof Kaufman. Tandem transformational game design: A game design process case study. In *Proceedings of the International Academic Conference on Meaningful Play*, 2016.

[517] Sigmund Tobias, J. Dexter Fletcher, and Alexander P. Wind. Game-Based Learning. In J. Michael Spector, M. David Merrill, Jan Elen, and M. J. Bishop, editors, *Handbook of Re-*

search on Educational Communications and Technology, pages 485–503. Springer New York, New York, NY, 2014.

[518] Tomorrow Corporation. *7 Billion Humans*. Game [Windows, macOS, Linux], 2018. Tomorrow Corporation.

[519] Gustavo F Tondello, Rita Orji, and Lennart E Nacke. Recommender systems for personalized gamification. In *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*, pages 425–430, New York, NY, USA, 2017. ACM, Association for Computing Machinery.

[520] Gustavo Fortes Tondello and Lennart E Nacke. A pilot study of a digital skill tree in gameful education. In *GamiLearn*, 2019.

[521] Keith Topping and Stewart Ehly. *Peer-assisted learning*. Routledge, 1998.

[522] Treasure. *Ikaruga*. Game [Windows], 2014. Treasure. Remake of the Japanese arcade machine *Ikaruga* (2001).

[523] Kathleen Tuite. Gwaps: Games with a problem. In *FDG*, 2014.

[524] Tunnel Vision Games. *Lightmatter*. Game [Windows], 2020. Aspyr.

[525] TV Tropes contributors. Golden path, 2020. Retrieved from `https://tvtropes.org/pmwiki/pmwiki.php/Main/GoldenPath` on January 5, 2021.

[526] Amos Tversky and Daniel Kahneman. Availability: A heuristic for judging frequency and probability. *Cognitive psychology*, 5(2):207–232, 1973.

[527] April Tyack and Elisa D Mekler. Self-determination theory in hci games research: Current uses and open questions. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–22, 2020.

[528] April Tyack, Peta Wyeth, and Madison Klarkowski. Video game selection procedures for experimental research. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–9, New York, NY, USA, 2018. Association for Computing Machinery.

[529] AlzheimersResearch UK. Sea hero quest vr - walkthrough video, 2017. Retrieved from `https://www.youtube.com/watch?v=CcM6Yu9d4pM` on December 2, 2021.

[530] University of Berlin. *Apetopia*. Game [Web Browser], 2013. http://apetopia.visual-computing.com/.

[531] University of Washington, Center for Game Science. *Foldit*. Game [Windows, macOS, Linux], 2008. https://fold.it/.

[532] University of Washington Center for Game Science. *Mozak*. Game [Web Browser], 2016. https://www.mozak.science/landing.

[533] U.S. General Services Administration. Eyewire — a game to crowdsource brain mapping, 2018.

[534] Krishna Teja Vaddepalli. Improving data quality in citizen science, 2019. Master's Thesis, LUT University School of Engineering Science.

[535] Valve. Portal. Game [Windows, macOS, Linux, PlayStation 3, Xbox 360, Android, Nintendo Switch], 2007. Valve Corporation.

[536] Jeroen JG Van Merriënboer. *Training complex cognitive skills: A four-component instructional design model for technical training*. Educational Technology, 1997.

[537] Jeroen JG Van Merriënboer, Otto Jelsma, and Fred GWC Paas. Training for reflective expertise: A four-component instructional design model for complex cognitive skills. *Educational Technology Research and Development*, 40(2):23–43, 1992.

[538] Jeroen JG Van Merriënboer and Liesbeth Kester. Whole-task models in education. *Handbook of research on educational communications and technology*, 3:441–456, 2008.

[539] Jeroen JG Van Merriënboer and Liesbeth Kester. *The four-component instructional design model: Multimedia principles in environments for complex learning.*, pages 104–148. Cambridge University Press, 2014.

[540] Jeroen JG Van Merriënboer and Paul A Kirschner. *Ten steps to complex learning: A systematic approach to four-component instructional design*. Routledge, 2017.

[541] Jeroen JG van Merriënboer and Paul A Kirschner. 4c/id in the context of instructional design and the learning sciences. *International handbook of the learning sciences*, pages 169–179, 2018.

[542] Jeroen JG Van Merriënboer and Dominique MA Sluijsmans. Toward a synthesis of cognitive load theory, four-component instructional design, and self-directed learning. *Educational Psychology Review*, 21(1):55–66, 2009.

[543] Maarten Vansteenkiste, Richard M Ryan, and Bart Soenens. Basic psychological need theory: Advancements, critical themes, and future directions, 2020.

[544] Carina Maria Veeckman, Sarah Talboom, Liesbeth Gijsel, Hilde Devoghel, and Annelies Duerinckx. *Communicatie bij burgerwetenschap: Een praktische handleiding voor communicatie en betrokkenheid bij citizen science*. SCIVIL, 2019.

[545] Jane Vella. *On teaching and learning: Putting the principles and practices of dialogue education into action*. John Wiley & Sons, 2007.

[546] Libuše Hannah Vepřek, Patricia Seymour, and Pietro Michelucci. Human computation requires and enables a new approach to ethical review. *arXiv preprint arXiv:2011.10754*, 2020.

[547] Libuše Hannah Vepřek. Multiplicities of meanings: Citizen science between knowledge production, gameplay and coping with everyday life. Presentation at the, Helsinki, Online. In *Proceedings of the 15th SIEF congress*, Helsinki, Online, June 2021.

[548] Katrin Vohland, Anne Land-Zandstra, Luigi Ceccaroni, Rob Lemmens, Josep Perelló, Marisa Ponti, Roeland Samson, and Katherin Wagenknecht, editors. *The Science of Citizen Science*. Springer International Publishing, Cham, 2021.

[549] Asher Vollmer. How to make great game tutorials, 2014.

[550] Luis von Ahn and Laura Dabbish. Designing games with a purpose. *Communications of the ACM*, 51(8):58–67, August 2008.

[551] Jérôme Waldispühl, Attila Szantner, Rob Knight, Sébastien Caisse, and Randy Pitchford. Leveling up citizen science. *Nature Biotechnology*, 38(10):1124–1126, 2020.

[552] Chih-Chien Wang and Yi-Shiu Chu. Harmonious passion and obsessive passion in playing online games. *Social Behavior and Personality: an international journal*, 35(7):997–1006, 2007.

[553] Shuhan Wang, Fang He, and Erik Andersen. A unified framework for knowledge assessment and progression analysis and design. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 937–948, 2017.

[554] Joe A Wasserman and Kevin Koban. Bugs on the brain: A mental model matching approach to cognitive skill acquisition in a strategy game. *Journal of Expertise/June*, 2(2), 2019.

[555] Etienne C Wenger and William M Snyder. Communities of practice: The organizational frontier. *Harvard business review*, 78(1):139–146, 2000.

[556] Sarah Elizabeth West and Rachel Mary Pateman. Recruiting and retaining participants in citizen science: What can be learned from the volunteering literature? *Citizen Science: Theory and Practice*, 2016.

[557] Andrew J Westphal, Anna L Butterworth, Christopher J Snead, Nahide Craig, David Anderson, Steven M Jones, Donald E Brownlee, Richard Farnsworth, and Michael E Zolensky. Stardust@ home: a massively distributed public search for interstellar dust in the stardust interstellar dust collector. *Lunar and Planetary Science XXXVI, Part 21*, 2005.

[558] Matthew M White. *Learn to Play: Designing Tutorials for Video Games*. CRC Press, Boca Raton, Florida, USA, 2014.

[559] Andrea Wiggins and Kevin Crowston. From conservation to crowdsourcing: A typology of citizen science. In *2011 44th Hawaii international conference on system sciences*, pages 1–10. IEEE, 2011.

[560] Andrea Wiggins and Kevin Crowston. Surveying the citizen science landscape. *First Monday*, 2015.

[561] Andrea Wiggins and John Wilbanks. The rise of citizen science in health and biomedical research. *The American Journal of Bioethics*, 19(8):3–14, 2019.

[562] G. Christopher Williams. Active learning: The pedagogy of the game tutorial, 2009.

[563] Chris Wilson. Jumping to conclusions - how to teach a player something they think they already know, 2017.

[564] Brian Winn and Carrie Heeter. Resolving conflicts in educational game design through playtesting. *Innovate: Journal of Online Education*, 3(2), 2006.

[565] Brian M Winn. The design, play, and experience framework. In *Handbook of research on effective electronic gaming in education*, pages 1010–1024. IGI Global, 2009.

[566] Jacob O Wobbrock, Leah Findlater, Darren Gergle, and James J Higgins. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 143–146, New York, NY, USA, 2011. ACM, Association for Computing Machinery.

[567] James Wootton. Decodoku: Quantum error rorrection as a simple puzzle game. In *APS March Meeting Abstracts*, volume 2017, pages V27–010, 2017.

[568] James R Wootton. Getting the public involved in quantum error correction. *arXiv preprint arXiv:1712.09649*, 2017.

[569] Olga A. Wudarczyk, Murat Kirtay, Anna K. Kuhlen, Rasha Abdel Rahman, John-Dylan Haynes, Verena V. Hafner, and Doris Pischedda. Bringing Together Robotics, Neuroscience, and Psychology: Lessons Learned From an Interdisciplinary Project. *Frontiers in Human Neuroscience*, 15:630789, March 2021.

[570] Rosa Yanez-Gomez, Daniel Cascado-Caballero, and Jose-Luis Sevillano. Academic methods for usability evaluation of serious games: a systematic review. *Multimedia Tools and Applications*, 76:5755–5784, 2017.

[571] YAW Studios. *War Solution — Casual Math Game*. Game [Windows], 2019. YAW Studios.

[572] Jonathan A Yee. Characterizing crowd participation and productivity of foldit through web scraping. Technical report, Naval Postgraduate School Monterey United States, 2016.

[573] Michael F Young, Stephen Slota, Andrew B Cutter, Gerard Jalette, Greg Mullin, Benedict Lai, Zeus Simeoni, Matthew Tran, and Mariya Yukhymenko. Our princess is in another castle: A review of trends in serious gaming for education. *Review of educational research*, 82(1):61–89, 2012.

[574] Zachtronics Industries. *SpaceChem*. Game [Windows, macOS, Linux, iOS, Android], January 2011. Zachtronics Industries, Washington, United States.

[575] Zachtronics Industries. *TIS-100*. Game [Windows, macOS, Linux], July 2015. Zachtronics Industries, Washington, United States.

[576] Zachtronics Industries. *Shenzhen I/O*. Game [Windows, macOS, Linux], 2016. Zachtronics Industries, Washington, United States.

[577] Zachtronics Industries. *Opus Magnum*. Game [Windows, macOS, Linux], 2017. Zachtronics Industries, Washington, United States.

[578] Binbin Zheng, Melissa Niiya, and Mark Warschauer. Wikis and collaborative learning in higher education. *Technology, Pedagogy and Education*, 24(3):357–374, 2015.

[579] John Zimmerman, Jodi Forlizzi, and Shelley Evenson. Research through design as a method for interaction design research in hci. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 493–502, 2007.

[580] Zooniverse, 2021. Retrieved from `https://www.zooniverse.org/` on December 2, 2021.

ProQuest Number: 30524406

INFORMATION TO ALL USERS
The quality and completeness of this reproduction is dependent on the quality
and completeness of the copy made available to ProQuest.

Distributed by ProQuest LLC (2023).
Copyright of the Dissertation is held by the Author unless otherwise noted.

This work may be used in accordance with the terms of the Creative Commons license
or other rights statement, as indicated in the copyright statement or in the metadata
associated with this work. Unless otherwise specified in the copyright statement
or the metadata, all rights are reserved by the copyright holder.

This work is protected against unauthorized copying under Title 17,
United States Code and other applicable copyright laws.

Microform Edition where available © ProQuest LLC. No reproduction or digitization
of the Microform Edition is authorized without permission of ProQuest LLC.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346 USA