How do Players and Developers of Citizen Science Games Conceptualize Skill Chains?

JOSH AARON MILLER, Northeastern University, USA
BRITTON HORN, Trinity University, USA
MATTHEW GUTHRIE, Northeastern University, USA
JONATHAN ROMANO, Eterna Massive Open Laboratory, State University of New York at Buffalo, Stanford University, USA
GUY GEVA, Eterna Massive Open Laboratory, Israel
CELIA DAVID, Princeton University, USA
AMY ROBINSON STERLING, Princeton University, USA
SETH COOPER, Northeastern University, USA

For citizen science games (CSGs) to be successful in advancing scientific research, they must effectively train players. Designing tutorials for training can be aided through developing a *skill chain* of required skills and their dependencies, but skill chain development is an intensive process. In this work, we hypothesized that free recall may be a simpler yet effective method of directly eliciting skill chains. We elicited 23 skill chains from players and developers and augmented our reflexive thematic analysis with 11 semi-structured interviews in order to determine how players and developers conceptualize skill trees and whether free recall can be used as an alternative to more resource-intensive cognitive task analyses. We provide three main contributions: (1) a comparison of skill chain conceptualizations between players and developers and across prior literature; (2) insights to the process of free recall in eliciting CSG skill chains; and (3) a preliminary toolkit of CSG skill-based design recommendations based on our findings. We conclude CSG developers should: give the big picture up front; embrace social learning and paratext use; reinforce the intended structure of knowledge; situate learning within applicable, meaningful contexts; design for discovery and self-reflection; and encourage practice and learning beyond the tutorial. Free recall was ineffective for determining a traditional skill chain but was able to elicit the core gameplay loops, tutorial overviews, and some expert insights.

CCS Concepts: • **Human-centered computing → Interaction design process and methods**; *Empirical studies in interaction design*; HCI design and evaluation methods.

Additional Key Words and Phrases: citizen science games; cognitive task analysis; skill tree; skill chain; skill decomposition

Authors' addresses: Josh Aaron Miller, miller.josh@northeastern.edu, Northeastern University, Boston, Massachusetts, USA; Britton Horn, bhorn@trinity.edu, Trinity University, San Antonio, Texas, USA; Matthew Guthrie, guthrie.mat@husky.neu.edu, Northeastern University, Boston, Massachusetts, USA; Jonathan Romano, jonathan@luxaritas.com, Eterna Massive Open Laboratory, State University of New York at Buffalo, Stanford University, USA; Guy Geva, guygeva2311@gmail.com, Eterna Massive Open Laboratory, Israel; Celia David, celia@eyewire.org, Princeton University, Princeton, New Jersey, USA; Amy Robinson Sterling, amyleerobinson@gmail.com, Princeton University, Princeton, New Jersey, USA; Seth Cooper, se.cooper@northeastern.edu, Northeastern University, Boston, Massachusetts, USA.

2573-0142/2021/9-ART244 $15.00
https://doi.org/10.1145/3474671

ACM Reference Format:
Josh Aaron Miller, Britton Horn, Matthew Guthrie, Jonathan Romano, Guy Geva, Celia David, Amy Robinson Sterling, and Seth Cooper. 2021. How do Players and Developers of Citizen Science Games Conceptualize Skill Chains?. *Proc. ACM Hum.-Comput. Interact.* 5, CHI PLAY, Article 244 (September 2021), 29 pages. https://doi.org/10.1145/3474671

1 INTRODUCTION

For citizen science games[1] (CSGs) to be successful in advancing scientific research, they must train and retain players via onboarding. Such training is often a non-trivial task. For example, Keep describes how the Eterna project evolved over several years in increasing complexity, such that new players "faced a considerable burden to engage meaningfully in the project" [48, p. 142]. Given the steep learning curves present in these games, better tutorials are needed [28]. However, in order to effectively train the players, developers first need to know what skills are actually required by the game so that they can design tutorials to teach these skills. Discovering the skills a game requires at an advanced level of play is also a non-trivial task, as players develop a professional vision to reach expertise in CSGs [58]. One useful tool to this goal is a skill chain, which models the hierarchical demands of the game's tasks [16].

But first, we require some 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.[2] *Skill chains*, on the other hand, were defined by game designer Dan Cook [16] who coined the term "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.[3] In this way, skill *trees* model the game state while skill *chain*s model the player.

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 can help developers identify breakdowns [43] 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 [69].

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.

[1]Games which crowdsource scientific advancement, see [18].
[2]Related to this meaning 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.
[3]Given the usage of "chain" as opposed to "tree," one might think that a chain is non-branching; however, this has not been the usage of the term in previous literature. For consistency, we also use the term "skill chain" to refer to the entire composition of player skills, despite the fact that it branches like a tree.

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 (CTA) proved to be an intensive task [39].

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 [13]. 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. For these reasons, we sought to test the efficacy of free recall in this context in the hopes that this method would be an easy but effective means of developing a skill chain.

While considering how players view the skill chain, we are 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?

This study has two goals. First, we aim 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 aim 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 address 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 CSGs: Foldit, Eterna, and Eyewire. We additionally member-checked [22] 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;[4] (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 are: (1) a comparison of skill chain conceptualizations between players and developers and across prior literature (i.e., the skill chain models of Cook [16] and Deterding [26]); (2) insights to the process of free recall in eliciting CSG skill chains; and (3) a preliminary toolkit of CSG skill-based design recommendations based on our findings.

2 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

[4]What You See Is All There Is

player design are beyond the scope of this paper, 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 [57]. 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.

2.1 Skill Learning in Citizen Science Games

Citizen science games are gameful or gamified applications that contribute to scientific research through the act of playing. This can include, for example, gamified collecting or classifying data, or games about solving novel, complex problems through human computation [73]. Most prior work on citizen science games has focused on data quality (e.g., [59, 67]) or participant motivations (e.g., [23, 44, 45]). Recently, however, researchers have been increasingly interested in the learning and expertise development that participants experience to solve these incredibly difficult problems [28, 48, 58]. Perhaps the earliest of this work was Andersen et al. [3] who investigated better tutorial design (to mixed results) for educational and citizen science games. Miller et al. [55] later built on this work to further examine tutorial design, concluding that the identification and development of player expertise is a key component of CSG design. Ponti et al. [58] use Goodwin's notion of a *professional vision* [34] to show how expert CSG players have developed new professional domains of expertise. Keep [48] studied Eterna longitudinally across several years, documenting its increase in complexity and emphasizing that knowledge organization, representation, and sharing are critical components of a successful CSG in order to onboard new players to an existing complex domain. Finally, Díaz et al. [28] surveyed CSG player experiences and found players expressing a lack of understanding, an acknowledgement of steep learning curves, and a need for better tutorials. Equally importantly, all of these factors co-occurred with discussion of engagement, often the losing of interest. In summary, recent work has begun uncovering the need for better training of CSG participants which remains an ongoing issue. Toward this goal, the present study explores the development of skill chains as a means to help designers create better tutorials by identifying which skills are prerequisites and common practices of expert play.

2.2 Cognitive Task Analysis in Games

Broadly speaking, this study is a Cognitive Task Analysis (CTA), of which there are many varieties of methods and purposes [21]. 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 [36]. 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).

2.3 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 [11]. Building on this "perceptual chunking" hypothesis, researchers have studied how expert and novice physics students create problem representations of physics problems [12]. 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 [2]. 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 [2]. 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 [2]. 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 [29] (and, on the other hand, learned by novices [36]), social and metagame aspects of expertise [27], learning curves and the habits of experts [41], and the markers of "extreme expertise" [52]. 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.

2.4 Skill Modeling

The skill chain model was originally proposed by game designer Dan Cook in 2007 [16]. Each *skill atom* consists 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" [16].

In 2015, Deterding built on Cook's model with the additional consideration of the player's motivation [26], 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 [40]. However, in the realm of player behavior modeling, skills can be actions, tactics, or strategies [6]. 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 [6].

Though beyond the scope of this paper, 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) [25, 33, 38]. This has been applied, for example, to automatically generate a partial ordering of practice problems for language learning [74]. 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" [20]. 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.

2.5 Applications of skill chains

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 [47, 56]. Kantharaju et al. [47] define the skill chain using the CTA methodology of Horn, Cooper, and Deterding [39] (discussed below) and define successful skill usage through binary behavior metrics, similar to the Behavlets approach [20]. For example, the skill "Testing before submitting" is defined by the behavior "Player tests before submitting."

Another study operationalizes Dan Cook's skill atom theory to implement Talin, a dynamic tutorial framework in the Unity game engine [5]. 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. [4, 35]). For this reason, Sarkar and Cooper [61] developed a disjoint skill model for simultaneously tracking game skills and task skills. Through this disjoint modeling, they were able to introduce players to overall more difficult tasks via more nuanced dynamic difficulty adjustment (DDA).

Lastly, skill chains have been applied within Foldit [40]. 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.

2.6 Designing skill chains

Most prior work on skill chains has assumed the developers can manually generate the skill chain. However, this process doesn't take into account empirical player behavior. Horn [38] 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*

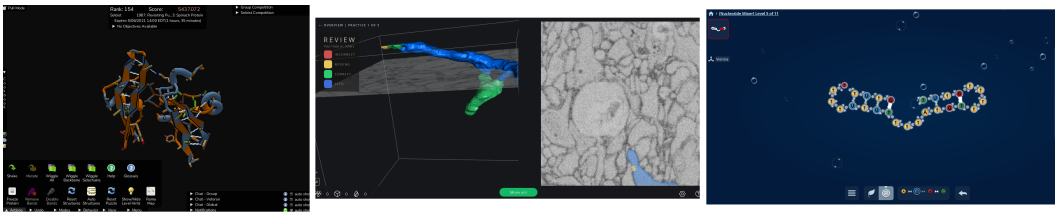

Fig. 1. Games studied: Foldit (Left), Eyewire (Center), and Eterna (Right). Screenshots taken by the first author.

> *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."*

To address this, Horn, Cooper, and Deterding attempted to elicit skill chains via (Skill-Based) Cognitive Task Analysis (*SB*CTA) [39]. 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" [39, p. 281]. Through analysis, the authors make six conclusions which are relevant and generalizable to the current study: (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.[5] Ultimately, the authors conclude that the CTA methodology produced something too raw and unstructured (Seth Cooper, personal communication, 2019), making it difficult to merge the results into a usable skill chain [39].

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. [39] 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" [17]. 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 [14]), 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.

[5] i.e., having many skills that share prerequisites rather than long chains of dependencies.

3 METHODS

The three games examined were Foldit[6] [19], Eterna[7] [50], and Eyewire[8] [68], chosen to be a representative sample of CSGs. Methods were approved by the appropriate Institutional Review Board.

3.1 Descriptions of Games Studied

Foldit is a 3D puzzle game where players attempt to fold a protein into a better shape in order to get a high score, through a combination of actions like shaking, wiggling, pulling, and running programmable scripts to combine these actions. Eterna has a very similar scientific goal, but is a 2D puzzle game. The primary actions of Eterna are mutating base pairs of RNA (sometimes by programmable scripts) in order to try to make the expected structure of the RNA match the desired target structure. Both Foldit and Eterna are driven by folding simulation engines to calculate the energy of the fold. Although not technically accurate, readers can think of these engines as similar to how the physics engines of commercial games simulate gravity in order to model platforming and other game interactions. Lastly, Eyewire is a 3D puzzle where players attempt to color in a 3D model of a neuron based on data from 2D slices. Success is determined based on other players' choice of coloring and an initial seed by Eyewire's task-assignment AI [51].

3.2 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 CSG player population as described by Curtis [24] in her synthesis of 13 studies containing demographics of CSG 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 CSG 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 [22], 11 participants were interviewed in a semi-structured format about their skill chain and play experiences.

[6]https://fold.it/
[7]https://eternagame.org/
[8]https://eyewire.org/

Fig. 2. 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 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.

3.3 Procedure

3.3.1 Skill Chains. Expert players and developers were referred to Cook's Gamasutra article[9] on skill chains (including his Tetris example) and an example skill chain of Mario (see Figure 2). Then they were asked to generate their own chain, such as by using the diagram-making tool *draw.io*,[10] 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.

3.3.2 Interviews. Experts, developers, and the Foldit novice recruited online were contacted for interviews as described in Section 3.2. 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.

3.3.3 Data Collected. As shown in Table 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.

3.4 Analysis

We first performed a multi-coder codebook thematic analysis [8, 10] on the 23 skill chains specifically to answer the research question "How do players and developers conceptualize the skills needed to play CSGs?". This work is theoretically framed from a constructionist perspective that assumes people create mental models to understand the world around them [63]. 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 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 [16, 26, 62].

For a preliminary, exploratory analysis, the generated skill chains were iteratively coded by three coders.[11] 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. [6]). 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

[9]https://www.gamasutra.com/view/feature/129948/the_chemistry_of_game_design.php
[10]https://app.diagrams.net/
[11]One skill chain was provided by the primary coder as a developer. The other two coders were neither players nor developers.

Braun and Clarke's reflexive methodology [8, 9], 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[12] are available at: https://osf.io/4evfk/.

4 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 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 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 [16] and Deterding [26].

4.1 Themes

Following a reflexive thematic analysis approach [8, 9], 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 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

[12]For privacy, please contact the first author for 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 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.

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*.

4.1.1 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

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 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 5.1.2).

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 [15] — 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[13] *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)

Part of this process involved *gaining an "eye"* or intuition for the decision-making, strategizing, and evaluation (cf. [58]). 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)

[13] A tool in Foldit that allows players to visualize and modify specific angles of the protein's fold.

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)[14]

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)

"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.

[14]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.

"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 [42] 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. [54]), 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 CSGs 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.

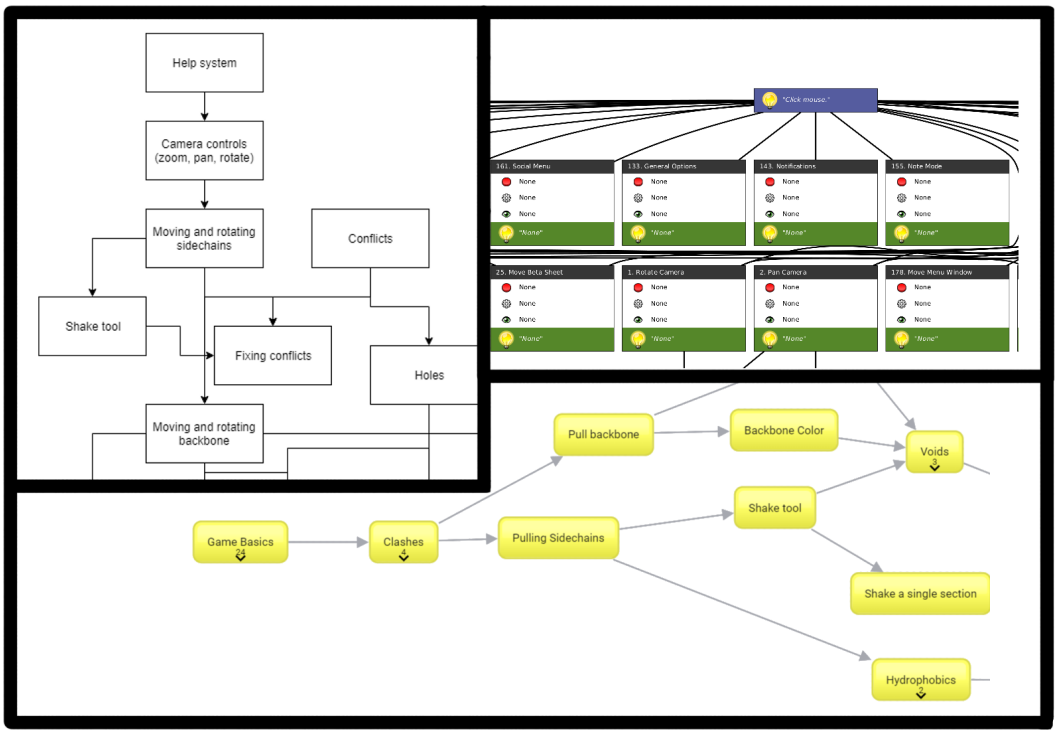

Fig. 3. Examples of describing the process of playing at a surface level. Top left: P3 (Foldit Novice) diagrams the camera controls and basic mechanics of Foldit. Top right: D2 (Foldit Developer) describes basic skills and game elements such as "Click mouse," "General Options", and "Rotate Camera." Bottom: 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.

4.1.2 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 3). 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).

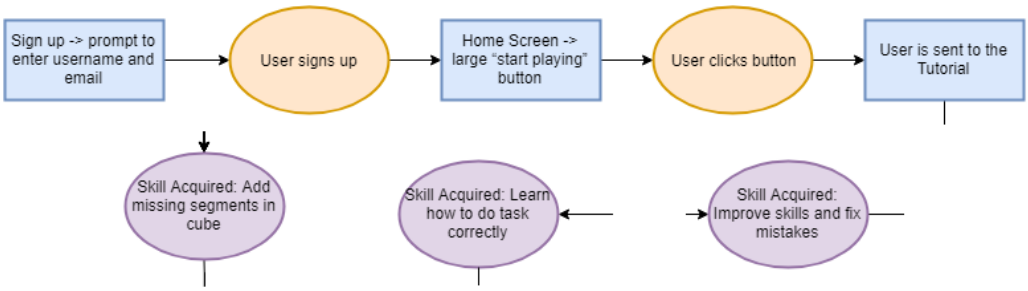

Fig. 4. Excerpts from D3 (Eyewire Developer) *equating the tutorial with the onboarding experience*. Top: the beginning of the chain procedurally walks through the tutorial experience. Bottom: the developer assumes that once a skill has been introduced, the player has acquired that skill. Note that this figure is made of excerpts from a skill chain created by a participant.

"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)

4.1.3 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 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 4 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 subtheme 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.

4.1.4 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[15] (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[16] 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 5.1.2, 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.

5 DISCUSSION

This study examined the methodology of directly eliciting skill chains from players and developers for CSGs. In the section below, we address our two research questions, the potential implications for design, and limitations of the study.

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 in the core gameplay loop. This agrees with the finding from Horn et al. [39] 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 [46], 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

[15]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.

[16]Other submissions included a plaintext list and a spreadsheet.

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. [39], 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 [1]). 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 [36], 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 [66]. Social learning has also been previously identified within Foldit by Bauer and Popović [7]. 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.

5.1.1 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 [47], though other more player-centric work represents this more gradually [5].

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 [39]: 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).

5.1.2 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 [16]. By Cook's model, skill chains are meant to capture the entirety of player learning and interaction for "pretty much any game imaginable" [16, 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 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 [71, 72].

5.2 RQ2: How effective is free recall as a method for directly eliciting the skill chain of a CSG 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. [39] 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 4.1.2, 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 [21, 37, 60], 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 [6]. Moreover, Horn et al. [39] 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., [5, 26, 40, 61, 69]) 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 [30, 31, 72]. 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.

5.2.1 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 [21].

5.3 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 Four-Component Instructional Design (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 [72].

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 [23, 44, 45], it supports skill practice through peer modeling and cognitive elaboration [49]. Moreover, both early success and early collaboration have been shown to correlate with increased participation [7]. 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 [65], 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 [49].

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 [32, 71].

Fifth, **design for discovery and self-reflection**. Discoveries trigger the generation effect [64], which promotes retention, and self-reflection promotes integration [53, 71, 72]. 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 [75].

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* [70] — 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.

5.4 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 CSGs. Moreover, we considered only four novice chains, which may not have been enough to reach theoretical saturation. On one hand, in Horn et al. [39], 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 [14, 21]. 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 CSG 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.

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 CSG 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 ACKNOWLEDGEMENTS

The authors would like to thank Fernando Portela for additional review and Samuel Wu-Ochs for help with the analysis. The authors thank the reviewers for their helpful feedback on earlier versions of this article.

REFERENCES

[1] Lada A Adamic. 2000. Zipf, power-laws, and pareto-a ranking tutorial. *Xerox Palo Alto Research Center, Palo Alto, CA* (2000).

[2] Vincent Aleven and Kenneth R Koedinger. 2013. Knowledge component (KC) approaches to learner modeling. *Design Recommendations for Intelligent Tutoring Systems* 1 (2013), 165–182.

[3] Erik Andersen, Eleanor O'Rourke, Yun-En Liu, Rich Snider, Jeff Lowdermilk, David Truong, Seth Cooper, and Zoran Popovic. 2012. The impact of tutorials on games of varying complexity. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 59–68.

[4] Sylvester Arnab, Theodore Lim, Maira B Carvalho, Francesco Bellotti, Sara De Freitas, Sandy Louchart, Neil Suttie, Riccardo Berta, and Alessandro De Gloria. 2015. Mapping learning and game mechanics for serious games analysis. *British Journal of Educational Technology* 46, 2 (2015), 391–411.

[5] Batu Aytemiz, Isaac Karth, Jesse Harder, Adam M Smith, and Jim Whitehead. 2018. Talin: A Framework for Dynamic Tutorials based on the Skill Atoms Theory. In *Fourteenth Artificial Intelligence and Interactive Digital Entertainment Conference*.

[6] Sander CJ Bakkes, Pieter HM Spronck, and Giel van Lankveld. 2012. Player behavioural modelling for video games. *Entertainment Computing* 3, 3 (2012), 71–79.

[7] Aaron Bauer and Zoran Popović. 2017. Collaborative problem solving in an open-ended scientific discovery game. *Proceedings of the ACM on human-computer interaction* 1, CSCW (2017), 1–21.

[8] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[9] Virginia Braun and Victoria Clarke. 2019. Reflecting on reflexive thematic analysis. *Qualitative Research in Sport, Exercise and Health* 11, 4 (2019), 589–597.

[10] Virginia Braun and Victoria Clarke. 2021. 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 (2021), 37–47.

[11] William G. Chase and Herbert A. Simon. 1973. Perception in chess. *Cognitive psychology* 4, 1 (1973), 55–81.

[12] Michelene TH Chi, Paul J Feltovich, and Robert Glaser. 1981. Categorization and representation of physics problems by experts and novices. *Cognitive science* 5, 2 (1981), 121–152.

[13] Richard Clark. 2014. Cognitive task analysis for expert-based instruction in healthcare. In *Handbook of research on educational communications and technology*. Springer, 541–551.

[14] Richard E Clark, Carla M Pugh, Kenneth A Yates, Kenji Inaba, Donald J Green, and Maura E Sullivan. 2012. The use of cognitive task analysis to improve instructional descriptions of procedures. *Journal of Surgical Research* 173, 1 (2012), e37–e42.

[15] Allan Collins, John Seely Brown, and Susan E Newman. 1988. Cognitive apprenticeship: Teaching the craft of reading, writing and mathematics. *Thinking: The Journal of Philosophy for Children* 8, 1 (1988), 2–10.

[16] Daniel Cook. 2007. The chemistry of game design. *Gamasutra*. http://www.gamasutra.com/view/feature/1524/the_chemistry_of_game_design.php. (2007).

[17] Nancy J Cooke. 1999. Knowledge elicitation. *Handbook of applied cognition* (1999), 479–509.

[18] Seth Cooper. 2014. *A framework for scientific discovery through video games.* Morgan & Claypool.
[19] Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popović, and others. 2010. Predicting protein structures with a multiplayer online game. *Nature* 466, 7307 (2010), 756–760.
[20] Benjamin Cowley and Darryl Charles. 2016. 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 (2016), 257–306.
[21] Beth Crandall, Gary Klein, Gary A Klein, Robert R Hoffman, and others. 2006. *Working minds: A practitioner's guide to cognitive task analysis.* Mit Press.
[22] John W. Creswell and Dana L. Miller. 2000. Determining validity in qualitative inquiry. *Theory Into Practice* 39, 3 (Aug. 2000), 124–130. DOI : http://dx.doi.org/10.1207/s15430421tip3903_2
[23] Vickie Curtis. 2015. Motivation to participate in an online citizen science game: A study of Foldit. *Science Communication* 37, 6 (2015), 723–746.
[24] Vickie Curtis. 2018. Who Takes Part in Online Citizen Science? In *Online Citizen Science and the Widening of Academia.* Springer, 45–68.
[25] Michel C. Desmarais and Ryan S.J.d. Baker. 2012. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction* 22, 1-2 (2012), 9–38.
[26] Sebastian Deterding. 2015. The lens of intrinsic skill atoms: A method for gameful design. *Human–Computer Interaction* 30, 3-4 (2015), 294–335.
[27] Scott Donaldson. 2017. Mechanics and metagame: Exploring binary expertise in League of Legends. *Games and Culture* 12, 5 (2017), 426–444.
[28] Carlos Díaz, Marisa Ponti, Pinja Haikka, Rajiv Basaiawmoit, and Jacob Sherson. 2020. More than data gatherers: exploring player experience in a citizen science game. *Qual User Exp* 5, 1 (Dec. 2020), 1. DOI : http://dx.doi.org/10.1007/s41233-019-0030-8
[29] Joey R Fanfarelli. 2018. Expertise in professional overwatch play. *International Journal of Gaming and Computer-Mediated Simulations (IJGCMS)* 10, 1 (2018), 1–22.
[30] Zach Gage. 2014. Designing for Problem-Solvers. PRACTICE: Game Design in Detail. (2014). Retrieved from http://stfj.net/DesigningForProblem-Solvers/ on May 13, 2021.
[31] Barry Garelick. 2013. Problem Solving: Moving from Routine to Nonroutine and Beyond. *Notices of the AMs* 60, 10 (2013).
[32] James Paul Gee. 2005. Good video games and good learning. In *Phi Kappa Phi Forum*, Vol. 85. The Honor Society of Phi Kappa Phi, 33.
[33] Carolina González, Juan C Burguillo, and Martín Llamas. 2006. A qualitative comparison of techniques for student modeling in intelligent tutoring systems. In *Proceedings. Frontiers in Education. 36th Annual Conference*. IEEE, 13–18.
[34] Charles Goodwin. 1994. Professional vision. *American Anthropologist* 96, 3 (1994), 606.
[35] M.P. Jacob Habgood and Shaaron E. Ainsworth. 2011. Motivating children to learn effectively: Exploring the value of intrinsic integration in educational games. *The Journal of the Learning Sciences* 20, 2 (2011), 169–206.
[36] Joseph Hesketh, Christoph Sebastian Deterding, and Jeremy Gow. 2020. 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.
[37] Robert R. Hoffman, John W. Coffey, Mary Jo Carnot, and Joseph D. Novak. 2002. An empirical comparison of methods for eliciting and modeling expert knowledge. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 46. SAGE Publications Sage CA: Los Angeles, CA, 482–486.
[38] Britton Horn. 2019. *A Skill-based Framework to Analyze the Generation and Presentation of Instructional Videogame Content.* Ph.D. Dissertation. Northeastern University.
[39] Britton Horn, Seth Cooper, and Sebastian Deterding. 2017. Adapting cognitive task analysis to elicit the skill chain of a game. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*. 277–289.
[40] Britton Horn, Josh Aaron Miller, Gillian Smith, and Seth Cooper. 2018. A Monte Carlo approach to skill-based automated playtesting. In *Fourteenth Artificial Intelligence and Interactive Digital Entertainment Conference*.
[41] Jeff Huang, Eddie Yan, Gifford Cheung, Nachiappan Nagappan, and Thomas Zimmermann. 2017. Master maker: Understanding gaming skill through practice and habit from gameplay behavior. *Topics in cognitive science* 9, 2 (2017), 437–466.
[42] Robin Hunicke, Marc LeBlanc, and Robert Zubek. 2004. MDA: A formal approach to game design and game research. In *Proceedings of the AAAI Workshop on Challenges in Game AI*, Vol. 4. 1722.
[43] Ioanna Iacovides, Anna L. Cox, Patrick McAndrew, James Aczel, and Eileen Scanlon. 2015. Game-play breakdowns and breakthroughs: exploring the relationship between action, understanding, and involvement. *Human–computer interaction* 30, 3-4 (2015), 202–231.
[44] Ioanna Iacovides, Charlene Jennett, Cassandra Cornish-Trestrail, and Anna L Cox. 2013. 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*. ACM, 1101–1106.

[45] Charlene Jennett, Laure Kloetzer, Daniel Schneider, Ioanna Iacovides, Anna Cox, Margaret Gold, Brian Fuchs, Alexandra Eveleigh, Kathleen Methieu, Zoya Ajani, and others. 2016. Motivations, learning and creativity in online citizen science. *Journal of Science Communication* 15, 3 (2016).

[46] Daniel Kahneman. 2011. *Thinking, fast and slow.* Macmillan.

[47] Pavan Kantharaju, Katelyn Alderfer, Jichen Zhu, Bruce Char, Brian Smith, and Santiago Ontanón. 2018. Tracing player knowledge in a parallel programming educational game. In *Fourteenth Artificial Intelligence and Interactive Digital Entertainment Conference*.

[48] Benjamin Ernest Keep. 2018. *Becoming Expert Problem Solvers: A Case Study in what Develops and how.* Ph.D. Dissertation. Stanford University.

[49] Diana Laurillard. 2013. *Teaching as a design science: Building pedagogical patterns for learning and technology.* Routledge.

[50] Jeehyung Lee, Wipapat Kladwang, Minjae Lee, Daniel Cantu, Martin Azizyan, Hanjoo Kim, Alex Limpaecher, Snehal Gaikwad, Sungroh Yoon, Adrien Treuille, and others. 2014. RNA design rules from a massive open laboratory. *Proceedings of the National Academy of Sciences* 111, 6 (2014), 2122–2127.

[51] Kisuk Lee. 2015. Crowd Intelligence in EyeWire. Technical report. (2015). Retrieved from http://kisuklee.wikidot.com/research#toc2 on May 6, 2021.

[52] John Lindstedt and Wayne Gray. 2013. Extreme expertise: Exploring expert behavior in Tetris. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 35.

[53] Tim Marsh. 2016. Slow serious games, interactions and play: Designing for positive and serious experience and reflection. *Entertainment computing* 14 (2016), 45–53.

[54] Josh Aaron Miller, Vivian Lee, Seth Cooper, and Magy Seif El-Nasr. 2019a. 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*. 535–542.

[55] Josh Aaron Miller, Uttkarsh Narayan, Matthew Hantsbarger, Seth Cooper, and Magy Seif El-Nasr. 2019b. 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*. 1–11.

[56] Huy Anh Nguyen, Xinying Hou, John Stamper, and Bruce M McLaren. 2020. Moving beyond test scores: analyzing the effectiveness of a digital learning game through learning analytics. In *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*. 487–495.

[57] David Osborne. 2016. Tales From a Crowdsourced Game Design. (2016). https://www.gdcvault.com/browse/gdc-16/play/1023216

[58] Marisa Ponti, Igor Stankovic, Wolmet Barendregt, Bruno Kestemont, and Lyn Bain. 2018. Chefs know more than just recipes: professional vision in a citizen science game. *Human Computation* 5 (2018), 1–12.

[59] Nathan Prestopnik, Kevin Crowston, and Jun Wang. 2014. Exploring data quality in games with a purpose. *iConference 2014 Proceedings* (2014).

[60] Ruziana Binti Mohamad Rasli, Faudziah Ahmad, and Siti Sakira Kamaruddin. 2014. A comparative study of conceptual graph and concept map. *Journal of Engineering and Applied Sciences* 9, 9 (2014), 1442–1446.

[61] Anurag Sarkar and Seth Cooper. 2019. Using a Disjoint Skill Model for Game and Task Difficulty in Human Computation Games. In *Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*. 661–669.

[62] Jesse Schell. 2008. *The Art of Game Design: A book of lenses.* CRC press.

[63] Dale H Schunk. 2012. *Learning theories an educational perspective sixth edition.* Pearson.

[64] Norman J Slamecka and Peter Graf. 1978. The generation effect: Delineation of a phenomenon. *Journal of experimental Psychology: Human learning and Memory* 4, 6 (1978), 592.

[65] Bethesda Softworks. 1994–2020. The Elder Scrolls. (1994–2020).

[66] Kurt Squire. 2012. Designed Cultures. In *Games, learning, and society: Learning and meaning in the digital age*, Constance Steinkuehler, Kurt Squire, and Sasha Barab (Eds.). Cambridge University Press, Chapter 2.

[67] Igor Stankovic, María Postigo Camps, Daniel Cuadrado Sánchez, Miguel A Luengo-Oroz, and Marisa Ponti. 2017. Performance of Players and Data Quality in a Citizen Science Casual Game. (2017). https://osf.io/preprints/socarxiv/vztxn/

[68] Ramine Tinati, Markus Luczak-Roesch, Elena Simperl, and Wendy Hall. 2017. An investigation of player motivations in Eyewire, a gamified citizen science project. *Computers in Human Behavior* 73 (2017), 527–540.

[69] Gustavo F Tondello and Lennart E Nacke. 2019. A Pilot Study of a Digital Skill Tree in Gameful Education. (2019).

[70] Valve. 2007. Portal. (2007).

[71] Jeroen JG Van Merriënboer, Otto Jelsma, and Fred GWC Paas. 1992. Training for reflective expertise: A four-component instructional design model for complex cognitive skills. *Educational Technology Research and Development* 40, 2 (1992),

23–43.
[72] Jeroen JG Van Merriënboer and Paul A Kirschner. 2017. *Ten steps to complex learning: A systematic approach to four-component instructional design.* Routledge.
[73] Luis Von Ahn. 2008. Human computation. In *2008 IEEE 24th international conference on data engineering*. IEEE, 1–2.
[74] Shuhan Wang, Fang He, and Erik Andersen. 2017. A unified framework for knowledge assessment and progression analysis and design. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 937–948.
[75] G. Christopher Williams. 2009. Active Learning: The Pedagogy of the Game Tutorial. (2009). https://www.popmatters.com/111485-active-learning-the-pedagogy-of-the-game-tutorial-2496084217.html

Received February 2021; revised June 2021; accepted July 2021