Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Computing and Information Systems | School of Computing and Information Systems

5-2022

Press A to jump: Design strategies for video game learnability

Lev PORETSKI

Anthony TANG
Singapore Management University, tonyt@smu.edu.sg

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research

 Part of the Information Security Commons

Citation

PORETSKI, Lev and TANG, Anthony. Press A to jump: Design strategies for video game learnability. (2022). *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, New Orleans, United States, April 29-May 5*.
Available at: https://ink.library.smu.edu.sg/sis_research/7892

This Conference Proceeding Article is brought to you for free and open access by the School of Computing and Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Computing and Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email cherylds@smu.edu.sg.

Press A to Jump: Design Strategies for Video Game Learnability

Lev Poretski
University of Toronto
lev.poretski@utoronto.ca

Anthony Tang
University of Toronto
tonytang@utoronto.ca

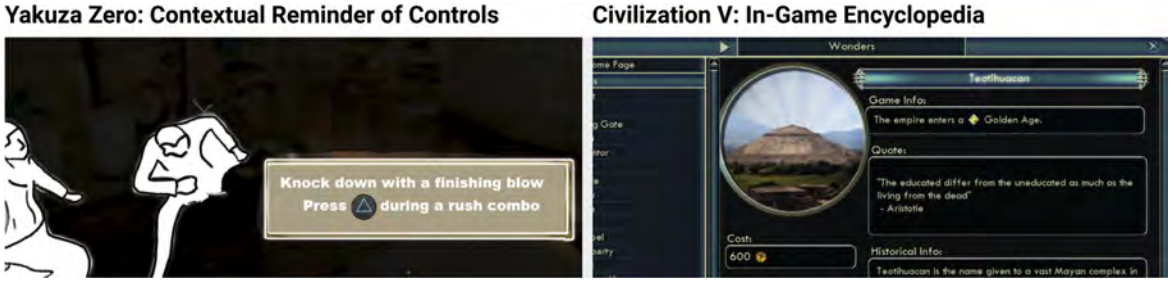

Figure 1: In the action video game Yakuza 0 (left), on-screen prompts instruct players on which buttons to press to execute the desired action. In the strategy game Civilization V (right), an in-game repository provides information about the game's controls and systems.

ABSTRACT

Learnability is a core aspect of software usability. Video games are not an exception, as game designers need to teach players how to play their creations. We analyzed 40 contemporary video games to identify how video games approach learning experiences. We found that games have advanced far beyond using simple tutorials or demonstration screens and adopt a range of repeatable and reusable design strategies using visual cues to facilitate learning. We provide a detailed descriptive framework of these design strategies, elucidating how and when they can be used, and describing how the visual cues are used to build them. Our research can be useful for both general HCI researchers and practitioners seeking to tap into the rich ideas from video game learnability design looking for practical solutions for their work.

CCS CONCEPTS

• **Human-centered computer**; • **Interaction design**; • **User centered design**;

KEYWORDS

Video games, learnability, visual cues, design strategies

ACM Reference Format:
Lev Poretski and Anthony Tang. 2022. Press A to Jump: Design Strategies for Video Game Learnability. In *CHI Conference on Human Factors in Computing Systems (CHI '22), April 29–May 05, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 26 pages. https://doi.org/10.1145/3491102.3517685

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CHI '22, April 29–May 05, 2022, New Orleans, LA, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9157-3/22/04…$15.00
https://doi.org/10.1145/3491102.3517685

1 INTRODUCTION

Because of increasing software complexity, designing products that are easy to learn and use is increasingly important [44]. HCI researchers and practitioners have formed a large body of knowledge on how to support learning in software products, where software learnability is a core aspect of usability assessment [46, 49]. Existing work has developed frameworks about supporting learnability (e.g. [31]), as well as documenting and proposing new ways to improve learnability in feature-rich software—e.g., through video guides [30], contextual tooltips and suggestions [36], or directly linking external video tutorials to the software [51].

In parallel, video games are growing in complexity and size: they now embody a vast range of control schemes, dynamic environments, and the scope of their experiences far exceeds the original arcade games of the 80s. Video games are distinct from feature-rich software because they occupy the intersection of storytelling, interaction, and entertainment; further, they employ various types of interaction and are comprised of different genres that each may require new things to be learned. Even more, players' motivations to engage with games are increasingly diverse [24, 65], which affects users' perceptions and attitudes toward learning. Consequently, designers have employed a wide range of tactics to support learnability, and it is not clear how these approaches can be effectively captured by current learnability frameworks [23]. For example:

1. In 1980, *PacMan* [L29] arcade machines would play a gameplay loop called "attract mode" when no players were playing. This would attract potential players and provide information about what to expect and how to control the game.
2. *Civilization* [L12], released in 1991, and its contemporary sequels, all provide the player with an in-game encyclopedia that provides information about how different elements of the game interact with one another.

3. As illustrated in Figure 1[1], in the action title *Yakuza 0* [L41], released in 2015, the game provides the player control prompts during gameplay to help players understand what buttons to press during intense battle sequences.

The problem is that our scholarly knowledge of supporting learnability in video games has not kept pace with the unique design language for learnability that has evolved in the video game design space. While existing work has identified problems with video game learnability [47], practical advice on how to solve these recurring problems has not emerged; consequently, designers may not know which learnability tactics and strategies are best applied in a given design situation [48]. For instance, some designers employ ways to teach the users game controls and mechanics information within the overall game narrative [18, 19, 47, 68] — an approach that is similar to incorporating learning into the gameplay itself [61]. Others have focused on making learning as enjoyable and engageable as the game itself [59]. Yet another approach emphasizes situated learning where failure and trial-and-error serve as powerful motivators and vehicles of learning [4].

Our work aims to identify and document ways that contemporary video game design supports learnability within the games themselves. To do this, we analyzed a sample of 40 highly rated mobile and console/PC games, identifying learning situations in the game, studying the context in which these situations arose, and describing a set of design strategies that designers have used to support learnability in video games. We identified a set of 11 common design strategies used in video games' design for learnability, we provide a detailed description based on their essence and function, for example: *Seeding in Cut-scene*, *Formal Documentation*, *Just-In-Time Reminders*, etc. These design strategies use several common lower-level design elements as the building blocks. We report on these elements as a multidimensional framework that describes how video game designers use these tactics as building blocks to help the player learn about the game. Specifically, we identify the dimensions of *Purpose*, *Format*, *Presentation Approach*, *Trigger*, *Constraints*, and *Repetitiveness*.

Our work makes two contributions. First, we identify 11 design strategies for video game learnability. Like in software engineering and UI design, researchers and practitioners can use these strategies as a reference, source of inspiration, and a baseline for supporting learnability in their games. Second, we articulate a framework of concrete design elements that video game designers use to execute on the design strategies. Thus, while we do not make claims of some strategies working better than others, and some elements supporting learnability more efficiently than others, we are able to create a library of current learnability practices that can be useful for future researchers wishing to explore how these practices benefit real players.

Based on our analysis, we illustrate how there may be opportunities for non-video game contexts to draw inspiration from these design strategies—particularly as software moves away from typical "desktop application" scenarios and into ubiquitous contexts.

[1]Some figures in this paper have been "photo-traced" from the original game to enhance the legibility of the in-game screen capture.

2 RELATED WORK

To set the stage for our work, we outline several past research threads around software learnability, describing this concept and research around it. We then review research exploring video game learnability, outlining the gap in the literature that this work addresses.

2.1 Learnability in Feature Rich Software vs. Video Games

Software learnability is the extent to which feature-rich software supports users in gaining *proficiency* [31] — the ability to operate the software with minimal "start-up" [46, 49], and *expertise*—the ability to use the advanced functionality of the software [21, 55]. Proficiency refers to users' ability to understand software workflows, to be aware of the software's functionality, to be able to locate and understand the functionality, and to eventually transition to expert behaviors [31]. Designing software for proficiency is challenging given the growing complexity of feature-rich software [3, 32, 37]. Conventional approaches to supporting learnability include extensive online documentation, implementation of automated wizards, and intelligent agents. Yet, studies demonstrate that users have difficulty gaining proficiency because of the effort required to seek tutorials, having to multitask between tutorial and software, and evaluating the usefulness of the tutorial before engaging with it [58]. Recent explorations have considered a variety of Q&A support structures around their products, incorporation of multimedia materials in learning tutorials [51], designing just-in-time tooltips that automatically trigger in certain contexts [30], and tutorials that actively guide the users through the interface [36] supporting reflection and pre-planning capabilities [3].

Video game researchers have considered learnability in related but distinct ways: usability [35], *intuitiveness* [6], *approachability* [17, 19], and *accessibility* [19, 56]. According to the ISO 9241-11 definition, *usability* refers to the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use [10]. Arguably, this definition may be too broad a term to describe design of learning processes. In turn, *intuitiveness* is defined as utilizing knowledge gained through experience with similar products or context [13], and, thus, can be too narrow to reflect the scope of all learnability aspects.

In this work, we specifically focus on the latter two concepts: *approachability* and *accessibility*, which reflect learning most specifically, and provide a comprehensive coverage of learnability context. Compared to the definition of learnability in feature-reach software, we view approachability and accessibility as aligned more closely with proficiency, which focus more on a player's early interactions with the game. In contrast, very little work investigated supporting expertise—what preliminary work exists suggests that achieving video game expertise has more to do with the time spent playing and motivation to master the game rather than specific design choices made by the game designers [40, 57].

Approachability refers to the extent to which it is easy for a novice to start playing a game without experiencing setbacks that discourage them from continuing to play [19]. Approachability is important for learnability design in video games [47] since video

game industry increasingly focuses on designing games for unforgiving novice audiences that demand optimal experience right from the start [16]. HCI literature has somewhat explored the concept of approachability, with researchers developing heuristic frameworks and usability procedures for approachability evaluation [19, 48]. However, approachability alone is not enough to capture the entire learning needs of the users. Where approachability is mainly concerned with the experience of novice players [68], games also need to support those who are already proficient in playing and want to advance even further [43]. Even more, approachability usually refers to the first 10-20 minutes of gameplay and emphasizes the task of keeping the player interested enough to continue after this initial amount of time [19, 47]. Thus, approachability cannot describe the learning aspects that need to support gameplay beyond the first 20 minutes: new and more advanced game controls, mechanics, and systems that gradually open themselves to the players.

Somewhat confusingly, video game scholars sometimes refer to these aspects using the term *accessibility*. Desurvire and colleagues define video game accessibility as "the factor by which the player can go from no knowledge of the game to adequate knowledge to be able to use the artifact to satisfaction" [19, 56]. Thus, in terms of the video game literature, accessibility may complement the idea of approachability and encompass the learning processes beyond the first minutes of gameplay. The definition of accessibility differs significantly from its contemporary use as support of users and communities with diverse physical and cognitive abilities (e.g., [11, 25, 52]). Thus, we should be cautious when defining accessibility simply in terms of learning aspects of video games.

For our purposes, then, we understand video game learnability as referring to all design aspects that help players with varying levels of game expertise to start playing the game without experiencing setbacks, acquire reasonable proficiency that allows them to continue playing successfully, and eventually achieve the level of satisfactory proficiency.

2.2 Research on How Video Games Support Learnability

Video game researchers have identified three high-level learnability game design principles focusing on general need for empowering the players, designing well-ordered problems, gradually presenting information and gameplay functionality [2], and applying system-thinking approach to design [61]. These aspects facilitate a pleasant and rewarding learning process resulting in players willingly engaging with long, complex, and difficult video games [27].

In practice, the high-level learnability principles often result in research work around studying video game tutorials (e.g., [4, 8, 29, 68]. Practitioners use tutorials to transmit information about the commands, the rules, the goals, and the behaviors expected by the player [8]. Researchers primarily distinguish internal tutorials (part of the software package itself) from external tutorials (situated outside the software as a separate resource) [1, 38, 41]. Internal tutorials in video games are often context-dependent [4, 9], appearing proactively when relevant to the specific content and situation within the game [4, 5, 27]. Additionally, prior work outlines the dimension of *freedom* in relation to the tutorial presentation, referring to whether the tutorial forces the user to perform the mechanics being described [4]. On the one hand, it is vital to give the players time and opportunities to experiment freely while learning new aspects of gameplay [54]. On the other hand, fully guided tutorials restricting the players only to the specific mechanics, actions, and functions, were previously found to be more effective in users' skills and knowledge acquisition [36].

Despite the proliferation of tutorials in video games, they are not always adequate in teaching players how to play [3]; game designers continue to explore new ways to teach players [28]. Several works described existing learnability design guidelines in general terms, such as using situated learning, reinforcement of skills, clear goals, and others, using individual examples of several contemporary video games (e.g., [19, 61]). For example, Desurvire and colleagues developed a set of heuristic guidelines for approachability that include practice of learned skills, providing information on demand, and defining a right challenge and balance of gameplay [19, 47]. Gee describe several contemporary games support learning, including providing information on-demand and just-in-time for the players [26–28].

Other studies analyzed larger game collections to identify game aspects that need learning, including game controls, mechanics, and story [8, 19]. Bizzocchi and colleagues stressed the importance of interface design in providing information to players, such as UI's "look and feel" based on behavioral mimicking and metaphors, consistency between interface and game world, and others [12]. Subsequently, we have seen a series of works focused on identifying interface cues, elements, and game processes related to learnability, such as awareness cues [69], geospatial information (maps) [66], and design of learning curves [42].

However, while current literature provides either a high-level overview of the learnability problem space or hyper-specific focus on certain design goals and activities (e.g., player awareness) and elements that can support these goals, we do not currently possess systematic practical knowledge on incorporation of learning processes throughout the game—particularly for learning beyond the first stages of a game. While there is some work that looks at UI cues in video games [66, 67, 69], researchers usually do not explicitly connect these cues to learnability, and there is no unifying framework that allows for standardization of learnability knowledge in game design. Consequently, academic research cannot not provide practical advice on many aspects of design for video game learnability [48].

Our work identifies multiple ways in which video games support learning. Building on existing research, we identify and describe design elements commonly used for learning in modern video games, developing a vocabulary to describe the learning cues. We also look at the larger context of when and how these elements are implemented to address specific learning situations and problems. We aim to support game design practitioners by providing clear descriptions of common design strategies that typical learnability challenges in games can be addressed. Our focus is on how games do this within the games themselves, reflecting game designers' intentions, rather than concerning ourselves with how communities use resources outside of the games (e.g., forums, wikis, etc.) to support learning.

Table 1: Categorization of the games in our sample according to their genre

Genre	Games N	Examples
Action/Adventure	23	The Last of Us 2 [L35]
Role Playing Games (RPG)	9	Persona 5 Royal [L30]
Simulation/Sport	7	EA NHL 2021 [L21]
Battle Royale/Battle Arena	6	Player Unknown's Battleground Mobile [L32]
Strategy	5	Hay Day [L26]
Arcade	5	Brawl Stars [L8]
Puzzle	5	Candy Crush [L11]

3 METHODOLOGY

The goal of the current work is to identify the in-game moments when games aim to teach players something about themselves, including how to control them, how to interact with different in-game systems, and how to advance toward the game destinations. We look at how different video games employ these notions, the common design language that facilitates learning experiences, and when specific design strategies are used within these games. We define a design strategy as a *common reusable pattern, practice, or solution used in video game design to teach users how to play the game.* Our inquiry is informed and guided by the previously identified theoretical concepts in the field of video game learning, particularly by the notions of video game *approachability* and *accessibility* [19]. In the next sections, we describe our data collection and analysis methodology.

3.1 Data Collection: Game Selection and Method

We identified the 60 most highly rated PC/console and mobile games of 2020, as featured on Metacritic and GameRanker web score aggregators at the start of February 1st, 2020, when we began our study. We limited our selection of video games by the current two generations of consoles (i.e., Xbox One and Series X, PlayStation 4 and 5, and Nintendo Switch), and well as PC games. In the case when a specific game had both console and PC versions featured on the Metacritic most rated 20 list, we chose the version with the higher rating. This sample captured a variety of platforms, genres, and game series. Beyond this base set, we also explored some games we had personal familiarity with or other games in game series (e.g., Assassin's Creed Valhalla [L7], Assassin's Creed Odyssey [L5], and Assassin's Creed Origins [L6]) or from the same or similar genre (e.g., first-person action, RPG, or strategy games). Our collection approach yielded games from a variety of genres presented in Table 1.

For each video game in our list, we watched 90 minutes of video game playthroughs on the video-sharing platform YouTube. We chose only the playthroughs that include only *audio-visual stream of gameplay* - with no external commentary from players and no interactions between the players and the audience of viewers[2]. We set a 90-minute period as a cut-off time as it was often the time where much of learning scaffolding was concentrated in games. To cover the material in a more systematic and balanced way, we decided to include one walkthrough per each game for our analysis. Given the video corpus and volume and variety of games, we decided to follow a consistent approach: one 90-minute video per game at the beginning of play. Our rationale for this consistent approach was to ensure a balanced view of the diverse design aspects of each title and game genre.

Additionally, we ensured that the game was started "from scratch", and the playthrough included all of the tutorials, cut-scenes, and introductory exposition information present in the game. Choosing such specific type of playthroughs provided us with the objective, highly-analyzable, and relatively-noiseless data focused on the specific object of study – visual, audial, and, potentially, haptic aspects of learnability interaction design within video games. At the same time, we did not distract ourselves by observations of player-specific performance on a gameplay process.

Finally, we played ten games by ourselves, choosing highly popular titles that include complex mechanics and system interactions, such as Doom: Eternal [L19], Animal Crossing: New Horizons [L2], and Persona 5 Royal [L30] to develop a first-hand understanding of users' learning needs during gameplay. We were previously acquainted with five games in this subsample, having played them in the past. However, we decided to replay these titles along with the new ones, this time paying particular attention to the learnability aspects of the games. For the games that we played by ourselves, we generally extended the procedure for 120 minutes instead of 90 minutes. We did this to account for having to repeat some sequences due to failure to complete them for the first time, increased amount of time required for screenshot capture, and increased period of learning, where we had to stop actively playing to read through the tutorials and understand the training sections.

At the initial stage of analysis, we identified learning situations or learnability elements in games. We define learning situations as moments during a game when it introduces the new gameplay aspects, systems, processes, and narrative embedding to the players and provides them with the information that allows them to understand and execute the required actions and act consciously and deliberately within the game world. When watching the game playthroughs, we took field notes and screenshots each time we identified a learning situation potentially taking place. As a result,

[2]Other types of videos were also considered but ultimately excluded. One type of video material we examined was tutorial videos where players/streamers explain how to perform certain in-game functionality or complete certain in-game challenge. We decided against this type of video in our analysis as it was often heavily dependent on the player's subjective performance and "meta-knowledge" of implicit in-game aspects that could be figured only by trial-and-error while playing the game, instead of being driven by the official in-game learnability design.

we created a database of 2434 screenshots, or 40 to 70 screenshots per game. We imported these materials into the qualitative analysis software and used them as a basis for subsequent analysis.

3.2 Analysis

We iteratively analyzed video recordings of the participants following the inductive qualitative thematic analysis guidelines by Braun and Clarke [14], aiming at creating detailed descriptions of the identified patterns within the qualitative data. At the first stage, we went over the collected data, identifying interesting and potentially relevant phenomena and creating open codes that describe this data. At the next stage, we categorized the created open codes into axial groups according to the relationships between them. At this stage, we started to notice recurring themes within the data that later served as a basis for describing the emerged design strategies. We adapted our process for multimedia material analysis following the recommendations by Suchman and Trigg [63]. Two researchers worked remotely to identify the codes within the data and iteratively refined coding schemes based on a study of the data.

We convened weekly to discuss our findings, identify any emerging discrepancies and disagreements, and arrived at a mutual agreement in our interpretation of the codebook. These meetings were captured on a large persistent online discussion board, where we collaboratively created the affinity diagrams reflecting our interpretations of the material and the emerging codes, organized relational schemata reflecting our brainstorming process, and subsequently outlined the emerging recurrent themes within our data. After the 3-month period of the analytic process, our discussion board contains more than 2400 individual elements that reflected our gradually evolving understanding of the video game learnability cues and design strategies. We continued until we had reached theoretical saturation—i.e., once we could describe every learnability element using our developed framework, and each added game did not add to significant new discoveries about learnability design.

Our analytic approach follows the thematic approach that others in the community have followed (e.g., [2, 45, 66, 69]). We initially focused on thematic analysis of game interfaces, collecting a catalog of video games, identifying relevant interface elements for learnability, and developing a framework of properties to describe and design these elements. We observed that designers frequently used certain combinations of learnability cues in common ways across different games, creating the recurring approaches in video game design for learnability. For example, one of the most prominent UI cues used in the game were visual tooltips that appeared on the screen for brief periods of time. We initially coded this element as a technical code [*UI: TOOLTIP*], subsequently reconceptualizing it as a [*REMINDER FOR ACTION*], and finally, producing a set of axial dimensions that describe application of this UI cue from every dimension: *[TRIGGER: SYSTEM-INITIATED]*, *[GOAL: INCREASE AWARENESS]*, *[MEDUIM: GRAPHIC/AUDIO]*, *[REPETITIVENESS: CONTEXTUAL]*, *[PRESENTATION: HUD]*.

To distinguish learnability cues from those that are simply a part of a game UI or gameplay processes, we specifically focused on those cues that exist to facilitate players' *understanding of the internal rules, patterns, and possible actions* within the game without being a part of the core gameplay mechanics. For example, if a red flashing dot appears over the enemy's head in a 3D stealth action game, a player will learn that enemies noticed them, and they need to perform some actions to resolve the situation (e.g., hide in a stack of tall grass, or climb on top of the building). Removing this red dot does not change the gameplay mechanics, because the player is still perfectly able to act as intended by game designers. However, this red dot enhances the player's understanding of how the game functions, which contributes to learnability of the product.

Although understanding learnability cues on a technical level was helpful, it wasn't enough for us to produce an in-depth understanding of higher-level video game learnability practices. After all, specific cues served as individual building blocks that helped to shape the usable, consistent, and well-recognized learnability design approaches. For example, contextual tooltips were often used in games to highlight interactive elements in graphically rich environments, or to remind players certain recommended actions from a wide range of possible options. We subsequently conceptualized this specific application of learnability cues by designers as a distinct design strategy – *Just-In-Time Reminder*. To formalize our understanding of these patterns, we re-watched video playthroughs of the games in our sample, focusing on answering the following questions:

- What are the common combinations of UI cues that video games use to provide learning?
- What gameplay situations lead to the application of which cues?
- Why did designers choose to apply a particular combination of cues instead of another?
- How do the identified designs support players' learning?

4 FINDINGS

We organize our findings in the following way: first, at a high level, where we describe the design strategies video game designers use to support learning before, during and after gameplay—these serve as high-level conceptual approaches that others can appropriate; second, we describe the building blocks of these strategies—the specific UI cues that are employed in video games that support and enact the design strategies. We describe these building blocks through a descriptive framework that builds on and extends the findings previously reported by Dillman and colleagues [20], specifically connecting these cues to enhancing learning for the players.

4.1 Design Strategies for Learnability

In our analysis we identified 11 distinct strategies that designers employ to address learning needs of the players in contemporary video games. We summarize the identified strategies in Table 2. Because we focus specifically on the in-game learning processes, the relationships between learning processes and the actual gameplay were important to us. Therefore, in our presentation of the design strategies for learnability, we organize them according to whether they were most frequently used before players start to play a game, during the gameplay, or after the play session.

Some of the strategies appeared most frequently in games of specific genres. For example, *The Invisible Hand* strategy was most common in the 3D action video games and used diegetic spatial cues

Table 2: Descriptive framework of design strategies for learnability

Game stage	Design Strategy	Short Description	Conditions
Before gameplay	Recaps	Recap of what had happened so far in the game	Long and complex games where players may return to after some time
	Seeding in the cutscene	Show glimpses of the relevant gameplay information in the opening cutscene/s	
	Assessing Prior Knowledge	Considering possible familiarity of the player with the game to partially or fully skip learning	A game combines new aspects on top of the aspects that exist in similar games or previous games in the series
	Tutorials	Explicitly teaching required functionality in well-defined stages using structured delivery of information	Existence of aspects that require explicit explanation and guidance to operate
During gameplay	The Invisible Hand	Diegetic cues directly in the game world	Large in-game environments where players need to be directed
	Practice in a Sandbox	A sandbox level following a learning sequence to practice the acquired skills	Mastery of learned skills is required for a subsequent play
	The Sixth Sense	Spatial non-diegetic cues on top of the game world	Complex graphically rich in-game environments where diegetic cues may be hard to notice
	Just-in-Time Reminder	Contextual tooltips in appropriate situations	Wide range of possible actions that are hard to remember
	Personal Advisor	An in-game agent that provides advice and recommends choices	Consequences of players' actions may emerge late in the game when it is no longer possible to reconsider one's decisions
After or adjacent to gameplay	Debriefing	Feedback on player's performance after the gameplay session	The game is structured around the short, well-defined replayable chunks of gameplay
	Documentation	Repository of interconnected articles with game information	Existence of many interconnected systems requiring formal descriptions for future reference

to direct players' attention to certain interactive elements within the environment. In contrast, the *Just-In-Time Reminders* strategy was almost universally applied throughout most video games in our sample, providing contextually appearing awareness cues during the appropriate gameplay situations. We mostly observed The *Recap* design strategy in large-scope video games that are heavy on in-game events, narrative arcs, and interaction possibilities, primarily in strategy and RPG genres. However, we observed that no game used learnability strategy exclusively. All of the learnability design aspects were driven by the specific contexts within different in-game situations and creativity of the overall design approach rather than being bound by the common tropes of certain genre. Rather, it was more a matter of the specific context within the different game situations and the overall design approach than specific genre tropes when selecting learnability strategies. Below we describe the strategies we identified and provide explanations regarding the learnability problems that these strategies intend to solve, conditions in which their implementation is appropriate, and a combination of cues used for their design.

4.2 Strategies for Learnability Before Gameplay

4.2.1 Recaps. **Goal.** If a player stops playing the game for some time and then decides to return to it, she can discover that she has forgotten much of the relevant gameplay information, or the state of the game systems may change significantly (in games where activity does not seize while player is logged out). Game designers need to brief returning players up so that they can remember how to play, learn new opportunities for interactions, and can easily continue.

Solution. To support returning players, some games have implemented recap sessions that provide short briefings for the players either each time the game is run or after a long layoff. The briefings may contain the recap of the narrative, a reminder of controls, and information about the state of the game systems that can help player understand how to use them effectively. For example, a popular mobile strategy game Clash of Clans [L15] presents an interstitial briefing screen with a detailed breakdown of the state of the game systems, including basic information about new elements that were built while player was away, if it detects that the player was absent for too long (Figure 5). The specific conditions after which games trigger the recaps may differ considerably, from appearing each time the game loads from the main menu (e.g., the Dragon Quest

Figure 2: Recap Design Strategy in the mobile strategy game Clash of Clans [L15]. If the game detects that the player was away for too long, it presents a recap with a state of the game systems.

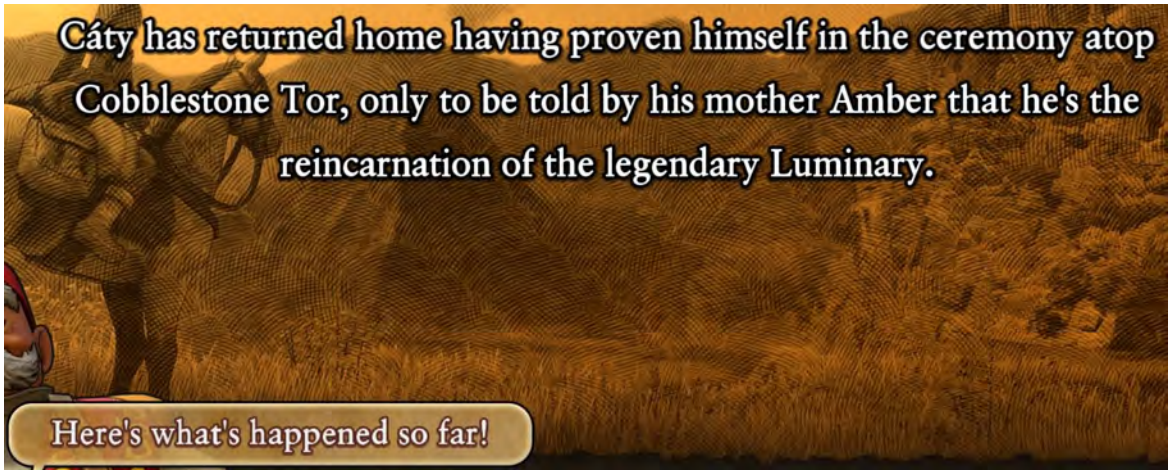

Figure 3: Recap Design Strategy in the RPG title Dragon Quest XI [L20]. At the start of each play session, players are reminded of the events that have happened so far.

XI [L20] in Figure 3) to being shown if significant changes have happened in the game world (e.g., Clash of Clans [L15] in Figure 2).

Conditions. It seems that this strategy can fit large-scale games that require multiple hours to complete and where multiple complex narratives and game system interactions gradually develop as the game progresses (all of the games where we observed this design strategy were very long and required more than 40 hours[3] to finish their main campaigns). In such games, there is a chance that players will put these games on hold at some point, only to return to them later. In this context, the Recap strategy is especially useful.

[3]According to the web aggregator Howlongtobeat.com that collects and weighs users' reports of time to complete video games

Figure 4: Seeding the information in the introductory cut-scene in Demon's Souls Remake [L18]. Players are vaguely shown the levels that they will encounter and some enemies that they will have to battle during their quest.

4.2.2 Seeding in the Cut-scene. **Goal.** Provide basic information about the story that motivates the gameplay right from the start of the game, so that even before players begin playing the game or stage, they have some understanding of what is to come.

Solution. Designers provided glimpses of gameplay and narrative in the introductory cut-scenes that appear before the game starts. Following the game narrative design terminology, we can define this strategy as "seeding" the information [62]: giving players a glimpse of what to expect in a very subtle way. Players can usually skip the cut-scenes. alternatively, the cut-scenes are presented only during the first run of the game.

Conditions. This design strategy is particularly well-suited for narratively rich games, particularly in RPG and Adventure genres, where understanding the game world, lore, and the story can significantly help players to advance in the game (e.g., players may have a better idea of where to go if they know what is happening in different regions in the game world). In Demon's Souls Remake for PlayStation 5 console [L18], opening cut-scenes show the player panoramic views of the places that they will visit and the enemies they will encounter during their journey (Figure 4). In the third person action-adventure God of War [L24], the opening cut-scene provides the initial exposition of the relationships between the main protagonist and his son, which will serve as a central narrative point throughout the whole game (Figure 5).

4.2.3 Assessing prior knowledge. **Goal.** Players may differ in the level of gaming experience and proficiency and may require different types and amount of learning information. Novice players will require more learning whereas more experienced players may wish to skip the introductory learning stages. Thus, it may be worth it to assess players' experience with similar video games at the start of the game and let the players choose the intensity of learning for themselves.

Solution. To assess prior knowledge of the player, certain games in our sample displayed a dialog window after players started it for the first time, asking them to choose whether they are new to the game series or type and want to engage in full learning experience, or they are experienced in similar games and want to view only cues related to the novel mechanics in this particular game. Sometimes players may skip the learning and tutorial stages completely.

Conditions. This strategy is particularly useful for games that are part of a series that follows the same design principles or games from the same genre that stick to the genre conventions closely. For example, in Total War: Three Kingdoms [L40], which is a part of a large game strategy series Total War, players are asked if they played the previous games in the series before? If yes, the game will only show information on new aspects unique to this installment. If no, the game will teach the players all aspects of interacting with the game (Figure 6).

4.2.4 Tutorials. **Goal.** Provide formally structured information about complex functionality, controls, mechanics, and interactions that require thorough breakdown.

Solution. Providing formal tutorials in the form of external UI windows or internal dialogs with non-player characters and other in-game objects (e.g., player may encounter the in-game information board with the detailed information about required functionality).

Conditions. Video games are highly interactive systems with many complex interfaces and interconnected modules. This is why we saw that tutorials, in one way or another, appear in almost all video games regardless of the game type. However, compared to other software, in video games, designers fine-tune tutorials for quick consumption: tutorials are often more structured, granular, and contextual, with each tutorial covering only a small part of the functionality. Many video game tutorials incorporate images,

Figure 5: Schematic representation of seeding the information in the introductory cut-scene in God of War [L21]. Players are introduced to the relationships between the main protagonist and his son, which serves as a central narrative point and contextualizes every event in the game.

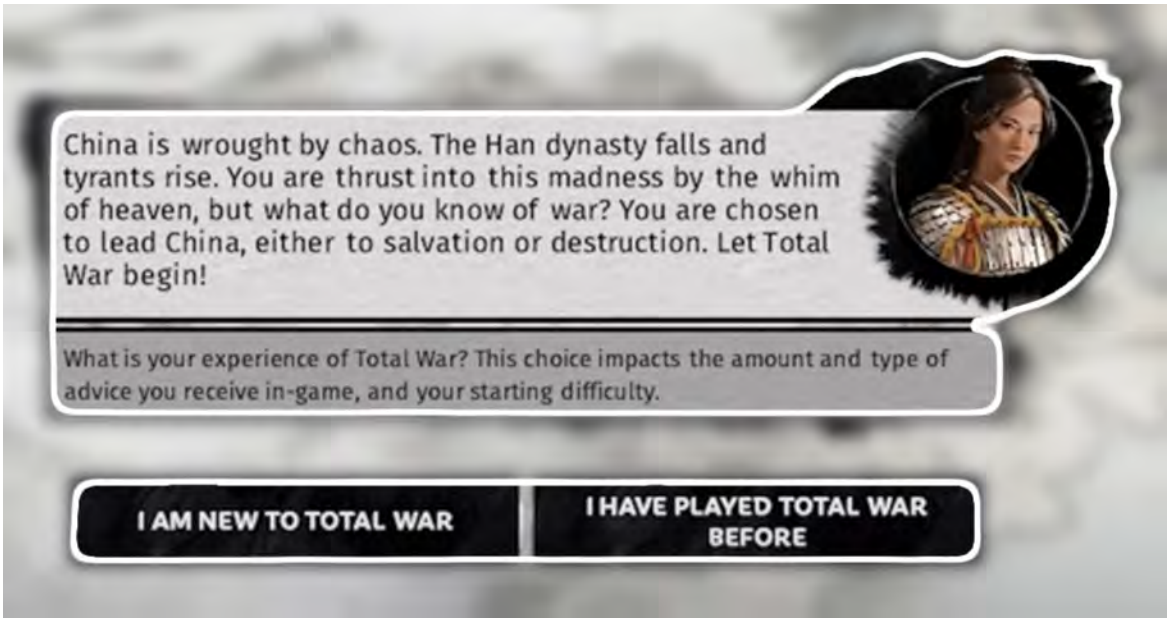

Figure 6: In the video game Total War: Three Kingdoms [L40], players can indicate whether they had played the previous games in the Total War series. If yes, the game will only provide learning on those aspects of the game that are unique to this installment.

Figure 7: Annotated screenshot with the example of The Invisible Hand strategy in The Rise of the Tomb Raider [L37].

video snapshots, text, and voiceovers. For example, in the first-person shooter Doom: Eternal [L19], the tutorials appear before the game requires the player to execute the functionality for the first time. The tutorial consists of a short video demonstrating this functionality accompanying the video with the textual description of how to execute the move.

4.3 Strategies for Learnability During Gameplay

4.3.1 The Invisible Hand. **Goal**: Subtle direction and guidance of the player by the game without breaking the player's immersion in the game world.

Solution: Designers incorporate subtle but distinguishable diegetic cues that organically blend into the game world's locations and situations and indicate points of interest to the player. In our sample, these cues were usually presented as visual cues; however, other modalities are also possible. For example, games can employ distinct audio cues in specific places in a game to indicate closeness of the player to a key location. Such cues are usually intended to support awareness of the possible actions, but not distract the player from paying attention to the game world. At the same time, game does not restrict players' control of their actions and decisions whether to follow the cues or not.

Conditions: We saw how designers apply this strategy in graphically rich games with explorable levels where players need to understand what elements of the environment are possible to interact with. In our sample, it was mostly 3rd- and 1st-person action games. Figure 7 demonstrates the example of the Invisible Hand strategy in a popular 3d action video game The Rise of the Tomb Raider [L37], where a white marking appears on surfaces that the hero can climb. The white marking supposedly represents surfaces that have been previously climbed by in-game enemies, indicating that our protagonist can climb them too. Another example is the interactive elements in the first-person game Resident Evil Village [L34], where many interactive elements are marked with yellow masking tape. When players see such elements, yellow highlighting makes it easy to understand that the items can be picked up or broken (Figure 8).

4.3.2 Practice in a Sandbox. **Goal**: Ensure that players have sufficiently mastered the necessary skills and have sufficient knowledge to advance further in the game.

Solution. When designers need to make players engage in learning the required functionality, they often implement practice levels that allow players to exercise learned skills and functions. Such levels usually directly follow the explicit learning stage and consist of the controlled interactive game environment that allows executing the just-learned aspects in a safe space. Usually, the game requires players to perform the learned functions or apply the new knowledge before allowing them to advance further.

Conditions. We saw the implementation of this strategy nearly universally among the video games in our sample. For example, complex and dynamic first-person shooter Doom Eternal [L19], uses sandbox environments each time new functionality is introduced (Figure 9). In a mobile shooter Call of Duty Mobile [L9], designers created first stages of the game as a series of lessons followed by

Figure 8: Annotated screenshot with the example of The Invisible Hand strategy in the game Resident Evil: Village [L34]. The interactive elements in the game are highlighted with yellow masking tape. Here, the lock is highlighted to indicate that a player can break it and open the locked door.

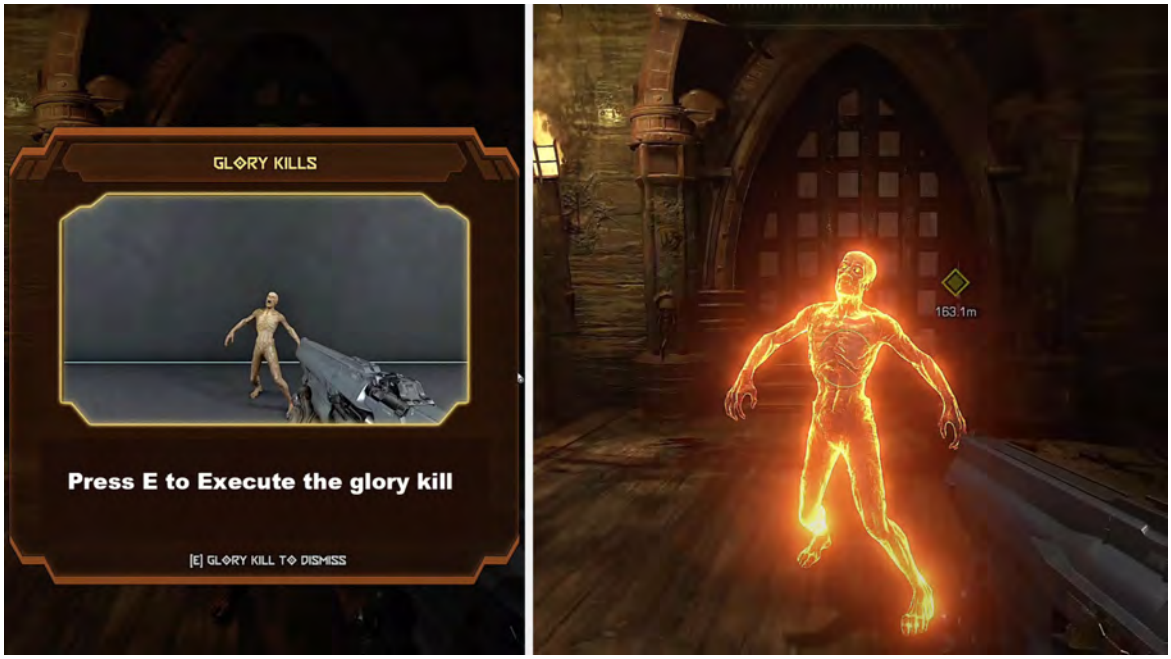

Figure 9: Annotated screenshot with the example of the Practice in a Sandbox design strategy in first-person shooter Doom Eternal [L19]. The player receives guidance on how to effectively dispatch demons (on the left) and proceeds to a sandbox environment with a conveniently located demon (on the right). The game then requires the player to repeat the learned skills before proceeding.

the mini-sandbox levels to practice the newly acquired knowledge (Figure 10).

4.3.3 The Sixth Sense. **Goal**: Provide players with information about the current state of the world, including the location of various points of interest and other interactive elements and agents

Figure 10: Screenshot with the example of the Practice in a Sandbox design strategy in first-person shooter Call of Duty Mobile [L9]. The player receives a lesson on basic controls and then proceeds to the sandbox level, where they can practice those skills.

in specific locations of the game world that players can miss or misinterpret if presented diegetically.

Solution: Many games provide non-diegetic overlays on top of the game world to indicate important elements of the environment, such as locations of the enemies, resources, and relevant places within the direct environment of the player. Usually, the designers give players the power to control when to trigger the appearance of this overlay and do not constrain players' motions and activities while the overlay is active. After players toggle The Sixth Sense, the overlay stays for a limited amount of time that is enough for the player to make sense of the environment and then disappears until players trigger it again. The gradual disappearance of the overlay allows to reduce the overabundance of spatial graphical elements extraneous to the game world, which may overload the presentation and distract the player from the diegetic level design elements.

Conditions: The application of The Sixth Sense design strategy makes most sense in first- and third-person actions games, where rich and graphically complex 3D environments make it hard for the players to identify all available interaction opportunities. Although presented via non-diegetic elements, designers often explain or justify the appearance of Sixth Sense in narrative terms. For example, open-world RPG *The Witcher 3 [L36]*, explains the player's ability to trigger this overlay with points of interest by the supernatural abilities of the main protagonist – namely, the ability to experience an incredible increase in perception (Figure 12).

Despite many similarities, The Sixth Sense's presentation is much more explicit than the Invisible Hand, allowing for greater flexibility and complexity of possible presented cues. For example, The Sixth Sense sometimes indicates elements within the environment that are hidden or obstructed by other objects. For example, in the stealth-action game *Assassin's Creed II* [L3], the application of The Sixth Sense allows the player to see the enemies that may be hidden by buildings or other elements in the environment, which is impossible to do via simple diegetic markers (Figure 11).

4.3.4 Just-In-Time Reminders. **Goal**. Provide players with in-situ instruction about the actions that are possible in specific gameplay situations.

Solution. Many games in our dataset display contextual prompts in specific game situations to recommend specific actions to the players either directly in the HUD or as spatially presented elements. Sometimes these prompts can include pictographic elements, such as button prompts, sometimes accompanied by the text labels, as well as meta-UI effects, such as flashing screens or vignetting edges of the screen. In our sample, Just-In-Time reminders are contextual. However, in certain cases, designers can choose to constantly display the instructions as a part of the in-game HUD if this information is relevant throughout the gameplay session. Just-In-Time Reminders primarily help to support players' awareness of possibilities and direct players' actions without explicitly constraining them to perform them. Implemented in such a way, Just-In-Time Reminders allow designers to directly connect the learning experience to the gameplay without breaking the immersion of the players.

Conditions. Similar to the Practice in a Sandbox strategy, Just-In-Time Reminders appear across the spectrum of game genres, from strategy games, such as Crusader Kings 3 [L17], 3D Action games like The Last of Us, part 2 [L35], and sports titles, such as EA NHL 2021 [L21]. Whenever a wide range of possible actions is available in the game, Just-In-Time Reminders indicate the possibility, availability, and desirability of specific actions. For example, in the stealth-action RPG game Assassin's Creed: Valhalla [L7], when the protagonist is hanging from the ceiling or a ledge, the Just-In-Time reminder pops up to remind a player how to climb down or move to another place (Figure 13). In the mobile video game Player Unknown's Battlegrounds [L32], the contextual window appears

Figure 11: Annotated screenshot with the example of The Sixth Sense design strategy in Assassin's Creed 2 [L3]. The player triggers the ability to see all the enemies in the vicinity as red-tinted shapes, and friendly NPCs as blue shapes (photo-traced for clarity), along with an indication of the enemies' direction and angle of view as the red arrow pointing from the enemies.

Figure 12: Annotated screenshot with the example of The Sixth Sense design strategy in The Witcher 3 [L36]. The player triggers the ability to highlight the enemies' tracks in the environment in red.

when the protagonist is near the items on the ground that she can pick up, to remind players of that possibility (Figure 14).

4.3.5 Personal Advisor. **Goal**. Inform the players of the future consequences of their actions at the earlier stages of the game, to increase players' understanding of how these consequences manifest themselves at the later stages of the game.

Solution. Designers may provide advice and wisdom to players. This advice is often given via anthropomorphized advisor characters that can be embodied and presented diegetically as explicit NPC characters within the world. Alternatively, they can appear as pop-ups or avatars on the screen each time players make an action that can have important consequences later in the game, to inform them of possible effects of their choices, and to provide momentary lessons about how the game functions.

Conditions. Because the role of Personal Advisors is to provide information on the future state of the world before this state becomes a game reality, this design strategy is uniquely fit for complex video games with a delayed feedback loop (e.g., turn-based

Figure 13: Annotated screenshot of Just-In-Time Reminder in the video game Assassin's Creed Valhalla [L7]. When the hero is in the position where he can execute the long jump down below, the contextual reminder momentarily appears in the HUD in the middle of the screen.

strategies and RPG). In such games, it is hard for inexperienced players to understand how their input produces ripple changes in the game systems and to anticipate the non-obvious consequences of these changes. For example, in turn-based strategy titles such as Civilization and Total War series, the personal advisor guides players to make the correct decisions each turn. The examples of the personal advisors in Civilization VI [L13] and Total War: Rome games [L39] are presented in Figure 15 and Figure 16, respectively.

4.4 Strategies for Learnability After or Adjacent to Gameplay

4.4.1 Debriefing. **Goal.** Provide feedback to the player, rating their performance across a range of predefined metrics (i.e., their expertise level on the current segment or round).

Solution. Games trigger Interstitial screens immediately after each gameplay session. During the debriefing, the game usually summarizes various information about players' performance and often assigns them a quantitative score/rating so that the players are aware of their demonstrated skill and mastery. The debriefing does not force players to repeat the gameplay sequence but leaves an option to replay it to achieve a better score.

Conditions. Games that use debriefings usually position them around well-defined replayable gameplay sequences or stages with a clear beginning and an end. Such stages are usually relatively short and self-contained, meaning that each session can be finished in one sitting. The stages are replayable, so players have the opportunity to improve their performance if their debriefing score is not high enough. For example, in a skating simulator Tony Hawk Pro Skater 1+2 Remake [L38], the player sees a debriefing screen with the details on their performance right after they finished a round of skating (Figure 17).

4.4.2 Documentation. **Goal.** Gather all in-game information into an accessible and searchable database, which players can access at any time to recall certain aspects of the game or to learn more about them in depth.

Solution. In the case of the particularly complex games, where players need to access the large amounts of knowledge about every aspect of the game, designers implement a wiki-like encyclopedia of interconnected pages that can be referenced from inside the game at any moment while playing. In less complex games, games usually implement simpler variations, such as a single window, sheet, or screen with learning information breakdown accessible at any time via the in-game menu.

Conditions. As in the case of game tutorials, it makes sense to store all learning information in an easily accessible form regardless of the game genre or type. Indeed, we observed this strategy in a wide variety of games, from turn-base strategies (e.g., Civilization series' in-game wiki - Civilopedia) to third-person action titles (in Tomb Raider: Rise of the Tomb Raider [L37], players can see all in-game controls by selecting the appropriate option in an in-game menu (Figure 18).

4.5 Descriptive Framework of UI Cues for Video Game Learnability

The designers enact the strategies we describe above through UI cues—effectively, the concrete building blocks used to realize the strategies. Based on the 2434 screenshots we collected from the 41 games in our sample, we developed a descriptive framework of UI cues used in learnability design with seven conceptual dimensions to describe each instance we observed. These instances cover a wide variety of multimedia modalities of the video game cues and UI elements, including textual prompts, voiceover narration during the gameplay, a combination of graphics, pictographic, and colors, as well as video and CGI sequences. Our analysis resulted in seven conceptual dimensions: *purpose, format, presentation approach, trigger, constraints, and repetitiveness* (summarized in Table 3). Each cue we identified was classified based on a level in each dimension. Below we describe each dimension in detail.

4.5.1 Purpose. Cues are meant to educate players about one or more aspects of the game. We distinguish between the three types of purposes:

- *Game Controls*, referring to the cues that provide an explicit description of the game controls. Figure 19a illustrates how Call of Duty: Mobile [L9] provides a description and demonstration of the basic game controls is displayed while the game loads,
- *Game Systems*, referring to the cues that help the player understand how the various systems and mechanics within the game work together and interact with each other. Figure 2b shows how Phasmophobia [L31] instructs the player that they will need to use the equipment to be able to locate and interact with in-game elements (Figure 19b),
- *Game Narrative*, referring to the cues that help players to understand various gameplay exposition elements and the overall in-game situation, comprehending the meaning and general directions of their actions within the game. In Pokémon Masters [L33], the NPC character hints at the player's

Figure 14: Annotated screenshot of Just-In-Time Reminder in Player Unknown's Battlegrounds Mobile [L32]. When the protagonist approaches the loot on the ground that they can pick up, the pop-up window appears on the screen to remind them to perform the action.

Figure 15: Personal Advisor in Civilization VI [L13]: personal advisor in the game recommends players to pay attention to developing resources on their territory.

teaming up with other characters to form a core of the in-game protagonists (Figure 19c).

4.5.2 Format. Cues provide a range of learning structures in terms of how learning information is provided and reinforced. In our observations, we saw the following types of cues' formats:

- *Awareness*, referring to the cues that serve to remind players about possible actions and options available to them during the gameplay while not explicitly requiring any action or reaction. Figure 20a illustrates how the action game The Last of Us Part II [L35] gives contextual prompts during fight sequences when the enemy intends to strike the player to remind players to dodge this attack.
- *Lesson*, referring to the structured information cues that are meant to explicitly teach players some aspects of the game outside of the main gameplay process (e.g., showing the information while pausing the game). Such cues clearly indicate the goal of the lesson beforehand and outline all required steps to achieve this goal. Figure 20b illustrates how the mobile battle arena game Mobile Legends [L27] pauses during the mission to explain how to use the on-screen joystick to move the protagonist.
- *Test*, referring to the cues that require the player to perform the mechanics or controls that are being taught in order to advance further in the game. Figure 3c illustrates how the mobile match-3 puzzle game Candy Crush L11] requires a player to match four yellow candies, only proceeding once the player has executed the functionality (Figure 20c).

Figure 16: Personal Advisor in global strategy game Total War: Rome Remastered [L39]: At the start of the game, the advisor introduces herself and explains to the players that she will guide them throughout the game's campaign.

Figure 17: Debriefing screen in Tony Hawk Pro Skater 1+2 Remake [L38]. After each round of skating, the player is shown the breakdown of their performance on several parameters, including the longest time spending doing specific skating tricks, like Grind, Manual, and Lip. The system also compares the player's current round with their best round so far, allowing them to understand how well they did this time.

4.5.3 Presentation Approach. Designers present and embed cues into games differently relative to the game experience and world. We observed the following four presentation approaches:

- *Interstitial*, where the cues appear on screens that are not a part of the gameplay (e.g., between levels). Figure 21a shows how the RPG title Dragon Quest XI [L20] tells the player what button to press on a separate black interstitial screen (Figure 21a).

Figure 18: The menu of all available controls accessible for reference at any time during the game in the title Tomb Raider: Rise of the Tomb Raider [L37].

Table 3: Descriptive framework of video game learning cues.

Dimension	Levels	Short Description
Purpose	Controls	How to control the game?
	Systems	How do different game systems interact?
	Narrative Elements	What is a player supposed to do and why?
Format	Awareness	Reminder about options during gameplay
	Lesson	Explicit teaching some aspect of the game
	Test	Requiring a specific action or learned functionality to advance
Presentation	Interstitial	Showing tips and hints on the screens that are not a part of core gameplay (e.g., loading screens)
	Head-Up Display (HUD)	Presenting cues within the game HUD
	Spatial	Aligning the cues with the game world
	Diegetic	Making cues a part of the game world
Trigger	System	The game triggers the cues
	Player-Implicit	Players trigger the cues by their actions within the game
	Player-Explicit	Players consciously trigger the cues
Constraints	System Lock	Locking all functionality except that, which is currently being explained
	Free Play	Players can freely play however they want
Repetitiveness	Constant	Cues are always present
	Contextual	Cues appear in certain situations
	One-Time	Cues appear one time during the game

- *Head-Up Display (HUD)*, where the cues appear as a part of the game's HUD during gameplay. Figure 21b shows how Dragon Quest XI [L20] provides a reminder that pressing the X button will open the game map is placed on top of the minimap and is a part of the game's UI.
- *Spatial Non-Diegetic*, where the cues align spatially with the game environment and update their presence with the changes in players' angle of view and relative position. Figure 21c shows how the mobile stealth-action game Assassin's Creed Identity [L4] shows a white line on the floor when the protagonist jumps to show the precise location where the jump will end.
- *Spatial Diegetic*, where the cues align spatially with the game environment and are a part of the game world. Figure 21d shows how the first-person shooter Half-Life 2 [L25] provides distinct color, outline, and label for which barrels will explode when shot.

4.5.4 Trigger. This dimension describes the proactiveness of the system in initiating the appearance of the cues to the player. We distinguish between the following trigger types:

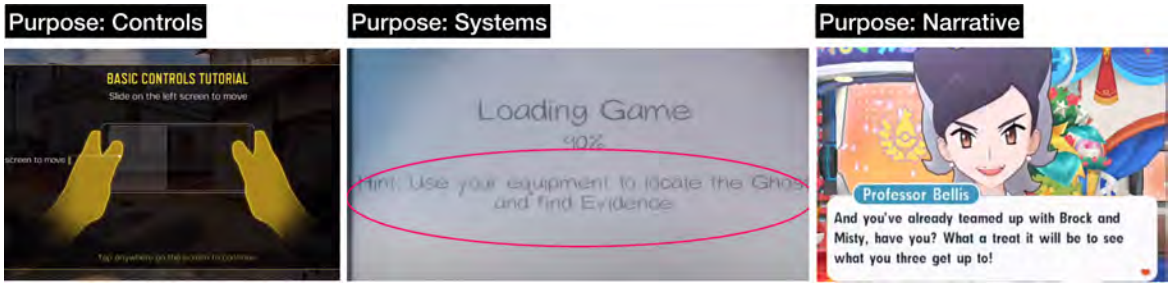

Figure 19: Screenshots illustrating different Purposes of learning cues in video games. (a): Call of Duty: Mobile [L9], the basic control scheme on the loading screen. (b): Phasmophobia [L31], the game instructs the player to use available equipment to interact with certain elements in the game – ghosts and evidence. (c): Pokémon Masters, an NPC in the game provides a narrative explanation of the player's battle team

Figure 20: Exemplary screenshots representing various forms of learning structure provided by learning cues in video games. (a): The Last of Us Part II [L35], when the enemy strikes the protagonist, the contextual prompt appears in this exact moment to remind the player that they can dodge it by pressing a certain button. (b): Mobile Legends [L27], the game explains how to control the protagonist using the on-screen joystick. (c): Candy Crush [L11], the game requires to match four candies in order to proceed further.

- *System*, describing the cues the system triggers directly without noticeable player interaction. For instance, in Brawl Stars [L8], when the player runs the game for the first time, the game runs a learning session.
- *Implicit-Player*, describing the cues that are implicitly triggered by certain players' actions within the game. For instance, in A Short Hike [L1], when the player gets closer to an NPC, this NPC explains to the player how to control flight.
- *Explicit-Player*, describing the cues that are explicitly triggered by players through a conscious choice or action. In Mortal Kombat 11 [L28], the player can choose to run the tutorial and training sessions from the main menu.

4.5.5 Constraints. This dimension describes whether the game constrains player actions when presenting learning information. We differentiate between the following two levels of constraints:

- *System Lock*: where the game restricts players' actions only to a certain subset of possibilities that it currently teaches. In Clash Royale [L16], during the first several missions, players are required to follow the game's instructions. They cannot press anywhere except where the system instructs them.
- *Free Play*: where the game does not restrict players' actions while it presents learning material. In Hay Day [L26], when the game teaches the player to harvest the crops, players can still freely roam around the environment and do other things.

4.5.6 Repetitiveness. This dimension describes how game repeats appearance of the cues throughout the game. We identified the following three types of repetitiveness:

- *Contextual*, cues repeatedly appear in certain situations within the game.
- *One-Time*, where cues appear only once and do not repeat. In Call of Duty: Modern Warfare [L10], the game teaches the player how to perform takedowns only once, during the first mission.
- *Constant*, where cues are constantly on the screen. In Fire Emblem: Three Houses [L22], during missions, the contextual prompts reminding the player how to control the characters are always present in the bottom-right part of the screen.

4.6 Comprehensiveness of the Framework

For each cue, we can use the dimensions outlined above to describe its relation to learnability. For instance, the cue in Figure 21d with the exploding barrel can be described with the following labels: Purpose: System, Format: Awareness, Presentation: Spatial Non-Diegetic, Trigger: Implicit-Player, Constraints: Free Play, and Repetitiveness: Contextual. Designers make use of a variety of different types of cues—often within the same game. Some are explicit

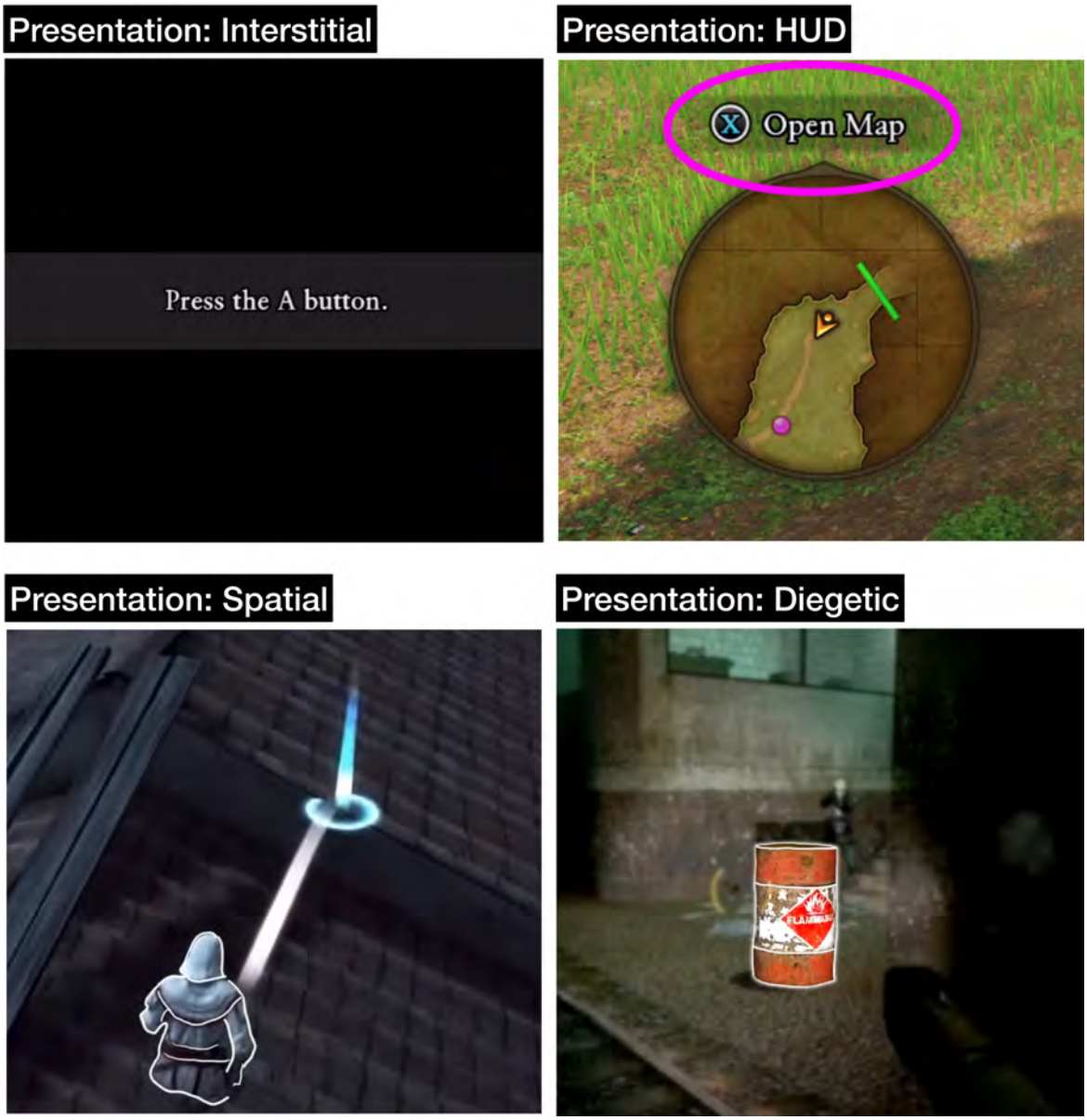

Figure 21: Exemplary screenshots representing various presentation of learning cues in video games. (a): Dragon Quest XI [L20], the instructions appear on the black screen outside of the gameplay sequence. (b): Dragon Quest XI [L20], the instruction on how open the in-game map appears as a part of the overall HUD. (c): Assassin's Creed Identity [L4], the white line shows the player the precise location of the jump. (d): Half-Life 2 [L25], the distinctly red explosive barrels stand out from the non-interactive environment.

in how they provide information to the player (e.g., through text), others appear as in-game, diegetic agents, while still others are far more visually or artistically subtle.

Similarly, designers use the cues in specific combinations to enact the design strategies described earlier. For example, the Sixth Sense strategy typically relies on Awareness cues presented Diegetically that are triggered Implicitly by the player, while enabling Free Play to provide the player with information about game Systems or Narrative. Practice in a Sandbox as a strategy typically uses a Test format, where the System triggers these, and Locks the gameplay—these last only One-Time, and are often about teaching the player about the Controls in the game.

5 DISCUSSION

Our work illustrates how video games have adapted learnability design approaches from feature-rich software and extended this

into a vibrant design language to support players' learning. We identify both high-level design strategies, as well as the UI building blocks these strategies use to provide learning. Below we describe and discuss the unique learning patterns that make the video game domain stand out from the rest of the software ecosystem. We then discuss the similarities and common design language that we noticed between video games and other software learnability strategies, and consider how we can harness the knowledge on the video game learnability for other applications.

5.1 What makes learning in video games so unique?

Video games are a special case of software. They drive forward the artistic vision of a designer, combining creative content and computer code in one holistic entertainment package. Moreover, video games are typically aimed at a diverse and wide audience of players, making accessibility a particularly important goal. This accessible, artistic, and entertainment character of video games means that the designers have different goals and objectives comparing to the designers of feature reach software, prioritizing enjoyment, empowerment, and inspiration as primary goals of their creation [18, 39, 50]. Consequently, video games often provide learning in a subtler fashion, situating it directly within the software itself, and learning stages go side-by-side with an actual gameplay experience. From the interaction perspective, we can describe video games as *mixed-initiative* interactive systems [33, 34] that attempt to predict and anticipate users' needs, goals, and attention states to provide learning information when relevant and desirable [34].

Nevertheless, the specific priorities of video game design result in unique tensions that each game designer needs to resolve and balance. In particular, the ability of the game to provide a sense of challenge and an intellectual reward in figuring out the rules of the game by oneself is one of the important motivators for many to play games [39, 45, 50]. Consequently, game designers should think about how to provide an easy learning experience without compromising players' sense of agency and independence. One way to subtly guide the players may be *The Invisible Hand* strategy, leaving the players to notice and interpret the subtly different environmental changes by themselves. Even the much more explicit *The Sixth Sense* strategy usually provides players with control when to trigger its appearance, allowing to decide when they to figure everything by themselves and when they want an additional aid from the game. A possible way to implement gradations in levels of challenge is to tie the appearance of specific learnability strategies to the level of difficulty that the players usually choose at the start of the game. It could be that on easy difficulty, *The Personal Advisor* appears to guide and direct the player, while on *hard,* players will be left to their own accord.

Another fundamental component of the video game experience is immersion [22, 53, 64]. Game designers often struggle with how to balance between the desire to immerse players into a game world and the necessity to teach players using extraneous non-diegetic information. Only one of the learnability strategies we observed is fully diegetic in nature (i.e., *The Invisible Hand*), and as we saw, by itself, it is often not enough to provide sufficient learning. One way in which the designers tackle this issue is by showing the UI elements only when they reflect some changes within the game environment or game parameters. For example, the ammo count in the first-person action game Resident Evil: Village [L34] shows only after the protagonist shoots or reloads the weapon.

Another unique challenge for video game designers is the necessity to accommodate different players' playstyles and motivations to engage with their video games. The existence of different video game players had previously been acknowledged by video game researchers [24, 53, 65]. Some players want the game to present strategic challenges, while others play to experience engaging narrative or a sense of relatedness [65], or competition [24]. Moreover, players' motivations may change over time. For example, players that started to play due to the in-game challenges may stay in the game because it provides them with social experiences [15]. Consequently, players may want to choose which learning strategies are particularly suitable for their experiences and change this setup whenever they desire through the in-game options menu, similarly to how the players interact with the third-person action game The Last of Us 2 [L35].

Finally, the complexity of video game environments that combine narrative, performative, and interactive elements presents the challenge of how to adapt style of learning to the in-the-moment in-game situations experienced by the players? Are they currently in the middle of the fighting sequence when presenting a fully blown tutorial is impossible? Are they about to encounter a new type of enemy that they need to learn about? Is the game currently loading, and we can show more structured information on the loading screen? The multiplicity of widely different situations that players experience within the scope of a single game results in need to combine all available learning strategies whenever relevant. In most video games that we saw, the designers used more than one strategy to provide an integrated learning experience. For instance, the game may seed introductory information in the cut-scene (*Seeding in the Cut-scene*), then present the formally structured breakdown of all controls during the loading sequence (*Formal Documentation*), highlight interactive elements within the level (*Invisible Hand*), present useful cues during a dynamic sequence (*Just-In-Time Reminders*), and debrief the player after she finishes the round (*Debriefing*).

Even though we focused on the most conventional and widely popular genres of games in our study, the clever and artful applications of an eclectic range of learnability strategies within every game raised our confidence about the overall generalizability of identified design strategies across different game types, genres, and devices. For example, we see no reason that design strategies that work in a sport simulation would not work in a racing game. The approaches that designers use in strategy games can arguably translate well to point-and-click adventure games, and RPG learnability design can work in rhythm-based arcades.

Of course, more niche game categories can serve as a source of rich knowledge and inspiration that we could not cover in our current game subset, and no strategy can be applied blindly, regardless of the particular context and goal of a designer. Rather, overall learnability strategy should reflect the designer's mindset and facilitate her latent dialog with her players, similarly to how writing style of a writer serves as an instrument of influencing the experience of book readers. Still, as the complexity of video

Figure 22: (a) player's view of their character when playing the sports title Tony Hawk Pro Skater Remake 1+2. Players may have trouble understanding that both the ramp to their left and the curb to their right can be used to perform skating tricks, as they appear as a part of the environment. (b) represents a hypothetical application of The Sixth Sense strategy for learnability, where players can trigger explicit indication of interactive surfaces and elements allowing them to understand what is usable or not during the gameplay.

games grows, the need for application of diverse range of learnability strategies grows with it, and we hope that our framework can provide a necessary scaffold for creative endeavor of future designers.

5.2 Exploring Potential Application of Design Strategies for Learnability in Games

To demonstrate how the learnability strategies could be applied on practice, consider the following example. In the sports video game Tony Hawk Pro Skater 1+2 Remake for PlayStation 5 game console, the players perform skating tricks and techniques while freely skating in open-world skating playgrounds. In the current design, players may experience difficulties in understanding what surfaces and elements in the environment are interactive and can be used to perform skating tricks. For example, in Figure 22a, players may have trouble understanding that they can use both the ramp to the left, and the curb to the right to perform various skating practices. Here, The Sixth Sense design strategy may prove itself useful. We demonstrate hypothetical application of The Sixth Sense strategy in Figure 22b. The ability of the player to highlight interactive elements can support players' ability to understand at a glance what elements they want to use in their play and plan accordingly.

Another possible example of how we can improve the existing interactions with games is a potential application of the *Assessing Prior Knowledge* strategy for learnability in mobile strategy Clash of Clans. Upon starting the game for the first time, the game typically guides a new player through a lengthy tutorial stage explaining in detail every possible interaction in the game. However, the game does not take into the account that experienced players sometimes start the game anew too, for example when changing their device or game account. In this case, experienced players would still need to go through the hurdles of doing every basic tutorial at the start of the new game. One of the possible solutions to this problem can be the incorporation of Assessing Prior Knowledge learnability strategy. At the start, the game can ask the players whether they are new or returning players. Based on the answer, the game can then choose to initiate the tutorial session or let the players skip the guidance altogether.

5.3 Video Game Learnability Design Strategies that Could Be Implemented in Other Contexts

Existing works have already started to look at video games as a source of practical knowledge for design of interactions beyond gaming, such as designing navigation for augmented reality [20]. We want to ask a similar question: what can we learn from video games that we can apply to learnability in other design fields, such as feature-rich software or even off-the-desktop experiences (e.g., AR interactions, ubiquitous computing)? Firstly, video games can serve as an example of learning being an inherent part of the overall software user experience. In games, the core processes of learning are always situated within the software, being inseparable from the actual gameplay and provided contextually when a situation requires it. Similarly, applications outside of the gaming context can interweave learning with the interaction itself, and provide momentary, contextual, non-restrictive cues throughout the whole experience. Secondly, video games emphasize that learning processes are more effective when players reinforce them with subsequent practice. This principle is useful in the context of many software experiences. For example, when watching a tutorial in Photoshop, the software can propose the user to perform the actions described in the previous step of the tutorial to advance it to the next step. Another way to enhance interaction is a possible application of *The Recap* design strategy. For instance, once a user opens the existing Photoshop project, the software can trigger a brief recap session, highlighting and explaining the elements that the user changed or added during the previous work session (Figure 23).

Another domain where a possible application of learnability strategies from video games can enhance user experience and performance may be augmented reality. Specifically, navigation and

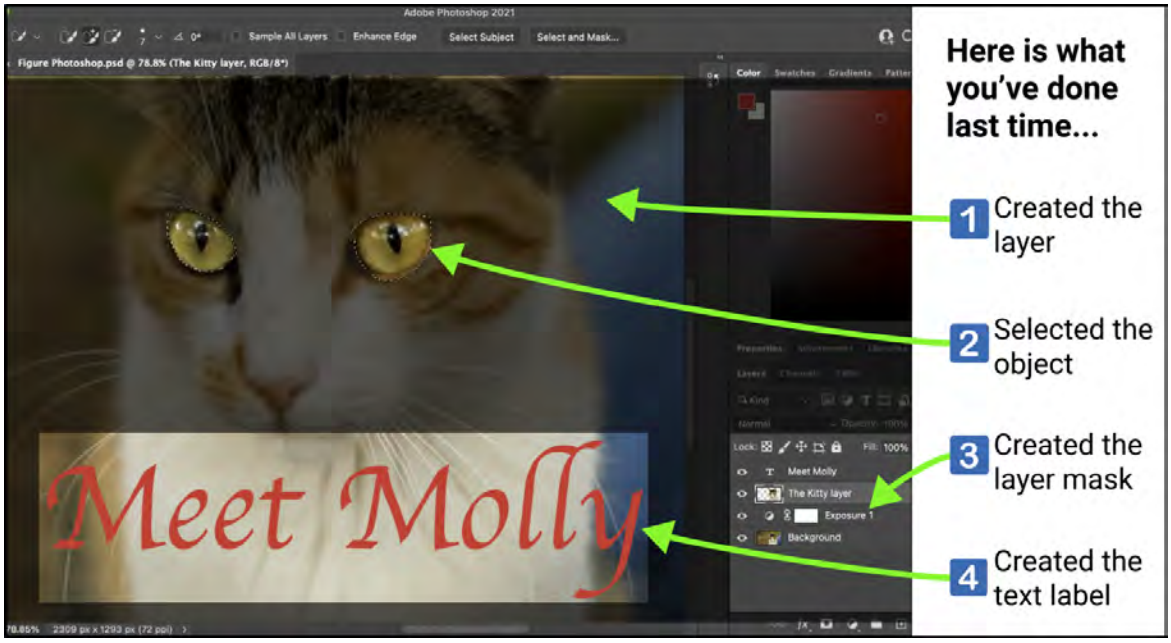

Figure 23: Imaginary implementation of The Recap design strategy in Photoshop. Once a user opens a project, the software provides a short recap, highlighting the elements that the designer worked on previously.

Figure 24: Imaginary application of the learnability design strategies in augmented reality navigation scenario: (a) A person uses an AR system with a common AR navigation interface design approach, where extraneous virtual elements show a person the location of the relevant shop in the shopping mall strip. (b) A person is equipped with a system that uses The Invisible Hand design strategy, where diegetic tracks on the floor point to the relevant location.

wayfinding are common use cases for AR, where the use of additional contextually aligned elements can aid users' wayfinding. Figure 24a is our schematic representation of such possible navigation scenario, where a user navigates to the specific shop in the shopping mall strip. However, extraneous virtual elements can often overload presentation, obscure relevant physical characteristics of the environment, and interfere with spatial cognition of users. *The Invisible Hand* design strategy can address this problem. For example, instead of introducing additional distinctive virtual elements, such as directional arrows and labels, the designers may simply choose to place the virtual tracks toward the entrance of the relevant store (Figure 24b).

To conclude, video game design can serve as a source of wisdom and examples for many design domains. Video games practitioners often act at the forefront of computing in their never-ending quest for technology-mediated inspiration and fun. Video game design practices, including those concerning design for learnability, are not an exception. Our experience with video games shows

that learning can be a creative, playful, artistic, and ultimately pleasant experience. It can enrich users' interactions with different systems rather than serve as a barrier to interaction. Even in sprawling strategic games, the intricacy of which reaches the levels comparable with feature-rich software, novice users can typically instantly start to play (as we experienced by ourselves), thanks to the implementation of learnability strategies that predict and guide users' workflows. We argue that such processes of background guidance and advice can enhance learning in many applications that are based on complex multi-step interactions, from industrial automation to interactions with smart devices. HCI practitioners will gain a lot from studying the video game domain and applying the accumulated knowledge for the ultimate benefit of their users.

5.4 Contributions

Our work contributes to the video game learnability research by demonstrating that video games have advanced far beyond simple tutorials and attract screens to make learning an enjoyable and engaging process. We show how the challenges that video game designers are facing result in the creation of increasingly original, granular, and contextualized designs of learning processes, and demonstrate this using a diverse sample of contemporary video games.

Our work's main theoretical contribution lies in developing a multidimensional learnability cues framework allowing us to form a sensitizing vocabulary for video game learnability research. We also add to the body of knowledge on video game design by describing the existing practical video game learnability design patterns. These should support future research on video games and help mediate the ongoing relationships between the theory and practice of video game design.

Previous research has already started to study how video games provide learning information. Several works focused on the provision of explicit tutorials that provide formal learning information to the players [4, 8, 29]. Additionally, the need for sandbox environments, which we acknowledged when describing the *Practice in a Sandbox* strategy, was previously discussed in the literature on video game learning [28, 54]. We were able to corroborate the findings, reinforcing them with the discovery that sandboxes were one of the universal vehicles for learning experiences present in virtually every game in our collection. The *Just-In-Time Reminders* pattern can be viewed in the context of what was defined as *just-in-time* and *proactiveness* dimensions in previous literature on game tutorial design [4, 5, 7].

Our work aligns with prior research exploring *awareness cues* in games. For example, Wuertz and colleagues [69], in their formulation of the design framework for *awareness cues* in games, draw on the same reasoning as we do when describing one of the dimensions for learnability cues ([*GOAL: AWARENESS*]. Toups and colleagues, in their research of video game signaling systems [67], characterize how games signal players about availability of certain actions and interactions in terms of being *player-perceived*, *avatar perceived*, and *meta-game*. This conceptualization is reminiscent of our taxonomy of cues' *presentation* dimension, with its distinction between *diegetic*, *spatially-aligned*, and *HUD* methods of presenting learnability elements.

We extend the existing knowledge by showing how video game designers and scholars can holistically view video game cues under the umbrella of learnability and creating the comprehensive descriptive framework describing such cues from every angle. We also show how games can meaningfully incorporate these cues throughout the game, not only for presentation of explicit tutorials but also for more dynamic and simultaneously consistent ways of reminding the player of all fragmented atomic bits of knowledge that they may require in specific gameplay occasions.

As a consequence, the research to date lacked a substantial empirical base on which to form the solid body of knowledge and could not provide practical advice on the identified learnability problems [47]. We can find certain parallels between the learnability strategies and cues identified in our study and other studies that described how interface design can support narrative goals of designers. For example, Bizzocchi and colleagues characterized the concept of "bridging" or "mixed-reality" interfaces, that are directly aligned onto the game world (similar to spatial non-diegetic cues used in what we define as *The Sixth Sense* strategy [12]. Our work continues to expand the video game design research by consolidating existing practical knowledge to describe concrete ways in which best practices of game design led to the specific solutions for game learnability. This will help practitioners in making informed design decisions around teaching players how to play their creations. Finally, we expect our findings to be useful to the HCI researchers and practitioners beyond the video game domain and expect that our description of the video game learnability practices can inspire novel perspectives on software interaction design.

5.5 Limitations, Future Work, and Conclusion

Limitations of Sample. Because we chose recent and most popular video games, our sample was somewhat skewed toward certain video game genres that typically attract the most audience, such as high-speed 3D action-adventure games and venerable RPG titles. Consequently, the more niche genres, such as racing, point-and-click, and 3D platformer games, were underrepresented in the current review. Each video game genre creates its own conventions and design knowledge [60] meaning that our list of identified design strategies is certainly non-exhaustive. However, observing how different video games mix-and-match learnability strategies, we can make an informed speculation that the patterns observed in the current work can (and probably are) applied beyond the types of games featured in our analysis. Nevertheless, at this point, we lack empirical evidence to definitely claim the fact, and future research should seek to include a more diverse representation of video games in the analysis.

Analytic Approach. We adopted the bottom-up approach to the analysis, identifying the specific learnability cues at the "atomic" level and then looking at how they comprise common design patterns. Future research should seek to create a better connection with the theoretical knowledge on game design to situate the identified patterns in overall game design principles. For example, it is interesting to connect the identified design patterns to the general principles identified by Gee, such as giving the players ability to explore and train in the sandboxed environments, designing clear and modular goals, and providing feedback on players' actions [47, 61].

"Natural Selection" Assumption vs. Evaluating Design Strategies. In our work, we make the implicit assumption that effective practices for learnability persist as the gaming landscape develops and evolves, however we have no direct evidence to this effect. It could be that among the design practices that we described in the current work, there are those that are less useful or effective in supporting players' learning. Thus, the next step in this line of research would be looking at the real players' practices and approaches to learning. Do they find certain strategies more useful? Are they enough to provide comprehensive learning? Do players consult outside sources of information or seek interpersonal help when stuck at the particularly challenging game sequence? How do they perceive the current state of the art in video game learnability design? Future researchers should interview and analyze the everyday practices of the real players to find the answer to these questions.

External learning sources. Finally, while our focus was on identifying specific ways in which learning is happening within games, future researchers may want to turn their attention on the practices around design and use of the external learning resources, such as wikis, forums, educational videos, and other forms of online content. External resources may be better suited to provide large amounts of complex structured information compared to the in-game learning approaches that, as we saw, often need to be brief, momentary, and increasingly contextualized. Consequently, while in-game learning may focus on merely making players proficient, external content may be directed toward those who wish to become experts. This is a hypothesis that is worth to consider in the context of a more general question – how do video game players acquire and develop expertise?

6 LUDOGRAPHY

[L1] A Short Hike, Adangryu, 2019, Microsoft windows, Linux, MacOS, Nintendo Switch

[L2] Animal Crossing: New Horizons, Nontendo, 2020, Nintendo Switch

[L3] Assassin's Creed 2, Ubisoft Montreal, 2009, Microsoft Windows, PlayStation 3, Xbox 360

[L4] Assassin's Creed Identity, Blue Byte, 2016, iOS, Android

[L5] Assassin's Creed Odyssey, Ubisoft Quebec, 2018, Microsoft Windows, PlayStation 4, Xbox One, Stadia

[L6] Assassin's Creed Origins, Ubisoft Montreal, 2017, Microsoft Windows, PlayStation 4, Xbox One, Stadia

[L7] Assassin's Creed Valhalla, Ubisoft Montreal, 2020, Microsoft Windows, PlayStation 4, Xbox One, PlayStation 5, Xbox Series, Stadia

[L8] Brawl Stars, SuperCell, 2017, iOS, Android

[L9] Call of Duty Mobile, TiMi Studio Group, 2019, iOS, Android

[L10] Call of Duty: Modern Warfare, Infinity Ward, 2019, Microsoft Windows, PlayStation 4, Xbox One

[L11] Candy Crush, King, 2012, iOS, Android, Microsoft Windows

[L12] Civilization, MIcroprose, 1991, Microsoft DOS

[L13] Civilization V, Firaxis Games, Microsoft Windows, OS X, Linux

[L14] Civilization VI, 2016, Microsoft Windows, macOS, Linux, iOS, Nintendo Switch, PlayStation 4, Xbox One, Android

[L15] Clash of Clans, Supercell, 2013, iOS, Android

[L16] Clash Royale, Supercell, 2016, iOS, Android

[L17] Crusader Kings 3, Paradox, 2020, Microsoft Windows, macOS, Linux

[L18] Demon's Souls Remake, Bluepoint Games, 2020, PlayStation 5

[L19] DOOM Eternal, id Software, 2020, Windows, PlayStation 4, Xbox One, Stadia, PlayStation 5, Xbox Series

[L20] Dragon Quest XI, Armor Project, Square Enix, 2017, PlayStation 4, Nintendo Switch, Nintendo 3DS, Xbox One, Microsoft Windows, Google Stadia

[L21] EA NHL 2021, EA Vancouver, 2020, Playstation 4, Xbox One, Playstation 5, Xbox Series

[L22] Fire Emblem: Three Houses, Intelligent Systems, Koei Tecmo, 2019, Nintendo Switch

[L23] Fortnite, Epic Games, 2017, Windows, macOS, Nintendo Switch, PlayStation 4, PlayStation 5, Xbox One, Xbox Series X/S, iOS, Android

[L24] God of War, Santa Monica Studio, 2018, PlayStation 4

[L25] Half-Life 2, Valve, 2004, Microsoft Windows

[L26] Hay Day, Supercell, 2012, Android, iOS

[L27] Mobile Legends, Moonton, 2016, iOS, Android

[L28] Mortal Kombat 11, NetherRealm Studios, 2019, PlayStation 4, PlayStation 5, Xbox One, Nintendo Switch, Xbox Series X and Series S, Microsoft Windows, Google Stadia

[L29] Pacman, Namco, 1980, Arcade

[L30] Persona 5 Royal, Atlus, P Studio, 2019, PlayStation 4

[L31] Phasmophobia, Kinetic Games, 2020, Microsoft Windows

[L32] Player Unknown's Battleground Mobile, PUBG Corporation, 2017, iOS, Android

[L33] Pokemon Masters, DENA CO, 2019, iOS, Android

[L34] Resident Evil: Village, Capcom, 2021, PlayStation 4, PlayStation 5, Xbox One, Xbox Series X and Series S, Microsoft Windows, Google Stadia

[L35] The Last of Us Part II, Naughty Dog, 2020, PlayStation 4

[L36] The Witcher 3, CD Projekt Red, 2015, Windows, PS4, Xbox One

[L37] Tomb Raider: The Rise of The Tomb Raider, Crystal Dynamics, 2015, PlayStation 4, Xbox One, PlayStation 5

[L38] Tony Hawk Pro Skater 1+2 Remake, Vicarious Visions, Neversoft, 2020, Nintendo Switch, PlayStation 4, PlayStation 5, Xbox One, Xbox Series X and Series S, Microsoft Windows

[L39] Total War: Rome Remastered, Creative Assembly, Feral Interactive, 2021, macOS, Microsoft Windows, Linux

[L40] Total War: Three Kingdoms, Creative Assembly, 2019, macOS, Microsoft Windows, Linux

[L41] Yakuza 0, Sega, 2017, PlayStation 4, Xbox One, Microsoft Windows, PlayStation 3

REFERENCES

[1] Ernest Adams. 2011. The designer's notebook: Eight ways to make a bad tutorial.

[2] Sultan A. Alharthi, Olaa Alsaedi, Z Toups, Theresa Jean Tanenbaum, and Jessica Hammer. 2018. Playing to Wait: A Taxonomy of Idle Games. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–15.

[3] Jessalyn Alvina, Andrea Bunt, Parmit K. Chilana, Sylvain Malacria, and Joanna McGrenere. 2020. Where is that Feature? Designing for Cross-Device Software Learnability. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference* (DIS '20), Association for Computing Machinery, New York, NY, USA, 1103–1115.

[4] 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* (CHI '12), Association for Computing Machinery, New York, NY, USA, 59–68.

[5] Daniel Atorf, Ehm Kannegieser, and Wolfgang Roller. 2019. Study on Enhancing Learnability of a Serious Game by Implementing a Pedagogical Agent. In *Games and Learning Alliance*, Springer International Publishing, 158–168.

[6] Benjamin Avery, Wayne Piekarski, James Warren, and Bruce H. Thomas. 2006. Evaluation of user satisfaction and learnability for outdoor augmented reality gaming. In *AUIC*.

[7] Batu Aytemiz. 2020. Towards Designing Out Helplessness: AI Interventions for Videogame Learnability. *AIIDE* 16, 1 (October 2020), 332–334.

[8] Fanny Barnabé and Björn-Olav Dozo. 2019. Narratological and rhetorical functions of video game tutorials: The case of NES games. (August 2019).

[9] Lawrence Bergman, Vittorio Castelli, Tessa Lau, and Daniel Oblinger. 2005. DocWizards: a system for authoring follow-me documentation wizards. In *Proceedings of the 18th annual ACM symposium on User interface software and technology* (UIST '05), Association for Computing Machinery, New York, NY, USA, 191–200.

[10] Nigel Bevan, Jim Carter, Jonathan Earthy, Thomas Geis, and Susan Harker. 2016. New ISO Standards for Usability, Usability Reports and Usability Measures. In *Human-Computer Interaction. Theory, Design, Development and Practice*, Springer International Publishing, 268–278.

[11] Kevin Bierre, Jonathan Chetwynd, Barrie Ellis, D. Michelle Hinn, Stephanie Ludi, and Thomas Westin. 2005. Game not over: Accessibility issues in video games. In *Proc. of the 3rd International Conference on Universal Access in Human-Computer Interaction*.

[12] Jim Bizzocchi, M. A. Ben Lin, and Theresa Jean Tanenbaum. 2011. Games, narrative and the design of interface. *International Journal of Arts and Technology* 4, 4 (2011), 460.

[13] Alethea Blackler, Vesna Popovic, and Douglas Mahar. 2003. The nature of intuitive use of products: an experimental approach. *Design Studies* 24, 6 (November 2003), 491–506.

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

[15] Stephen Carradini and Anya Hommadova Lu. 2020. New motivations: Change over time in motivations for mobile gaming. *Journal of Gaming & Virtual Worlds* 12, 3 (2020), 259–285.

[16] Heather Desurvire, Kalle Jegers, and Charlotte Wiberg. 2007. Evaluating fun and entertainment : Developing a conceptual framework design of evaluation methods. In *INTERACT'07*.

[17] Heather Desurvire and Charlotte Wiberg. 2008. Master of the game: assessing approachability in future game design. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '08), Association for Computing Machinery, New York, NY, USA, 3177–3182.

[18] Heather Desurvire and Charlotte Wiberg. 2009. Game Usability Heuristics (PLAY) for Evaluating and Designing Better Games: The Next Iteration. In *Online Communities and Social Computing*, Springer Berlin Heidelberg, 557–566.

[19] Heather Desurvire and Charlotte Wiberg. 2015. User Experience Design for Inexperienced Gamers: GAP—Game Approachability Principles. In *Game User Experience Evaluation*, Regina Bernhaupt (ed.). Springer International Publishing, Cham, 169–186.

[20] Kody Dillman, Terrance Mok, Anthony Tang, Lora Oehlberg, and Alex Mitchell. 2018. A visual interaction cue framework from video game environments for augmented reality. *Conference on Human Factors in Computing Systems - Proceedings* 2018-April, (2018), 1–12.

[21] Alan Dix, Alan John Dix, Janet Finlay, Gregory D. Abowd, and Russell Beale. 2003. *Human-computer interaction*. Pearson Education.

[22] Laura Ermi. 2005. Fundamental components of the gameplay experience: Analysing immersion. (2005).

[23] Adrian Fernandez, Emilio Insfran, Silvia Abrahão, José Ángel Carsí, and Emanuel Montero. 2012. Integrating Usability Evaluation into Model-Driven Video Game Development. In *Human-Centered Software Engineering*, Springer Berlin Heidelberg, 307–314.

[24] Quantic Foundry. 2016. Gamer Motivation Model.

[25] Lee Garber. 2013. Game accessibility: enabling everyone to play. *Computer* 46, 06 (2013), 14–18.

[26] James Paul Gee. 2003. What video games have to teach us about learning and literacy. *Computers in Entertainment* 1, 1 (2003), 20.

[27] James Paul Gee. 2004. Learning by design: Games as learning machines. *Interactive educational multimedia: IEM* (2004), 15–23.

[28] James Paul Gee. 2005. Learning by Design: Good Video Games as Learning Machines. *E-Learning and Digital Media* 2, 1 (March 2005), 5–16.

[29] Michael Green, Ahmed Khalifa, Gabriella Barros, and Julian Togellius. 2017. "press space to fire": Automatic video game tutorial generation. *AIIDE* 13, 2 (2017), 75–80.

[30] Tovi Grossman and George Fitzmaurice. 2010. ToolClips: an investigation of contextual video assistance for functionality understanding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10), Association for Computing Machinery, New York, NY, USA, 1515–1524.

[31] Tovi Grossman, George Fitzmaurice, and Ramtin Attar. 2009. A survey of software learnability: metrics, methodologies and guidelines. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '09), Association for Computing Machinery, New York, NY, USA, 649–658.

[32] Kathy Haramundanis. 2001. Learnability in information design. In *Proceedings of the 19th annual international conference on Computer documentation* (SIGDOC '01), Association for Computing Machinery, New York, NY, USA, 7–11.

[33] Marti A. Hearst, J. Allen, Eric Horvitz, and C. Guinn. 1999. Mixed-initiative interaction. *IEEE Intell. Syst.* 14, 5 (1999), 14–23.

[34] Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (CHI '99), Association for Computing Machinery, New York, NY, USA, 159–166.

[35] Jacqueline Hung. 2015. Usability and learnability improvements for the TaleBlazer game editor. Massachusetts Institute of Technology.

[36] Caitlin Kelleher and Randy Pausch. 2005. Stencils-based tutorials: design and evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '05), Association for Computing Machinery, New York, NY, USA, 541–550.

[37] Kimia Kiani, Parmit Chilana, Andrea Bunt, Tovi Grossman, and George Fitzmaurice. 2020. "I Would Just Ask Someone": Learning Feature-Rich Design Software in the Modern Workplace. In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 1–10.

[38] Kimia Kiani, George Cui, Andrea Bunt, Joanna McGrenere, and Parmit K. Chilana. 2019. Beyond" One-Size-Fits-All" Understanding the Diversity in How Software Newcomers Discover and Make Use of Help Resources. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, dl.acm.org, 1–14.

[39] Julian Kuklich. 2005. Precarious Playbour: Modders and the Digital Games Industry. *The Fibreculture Journal* 05, 05 (2005).

[40] Andrew Latham, Lucy Patston, and Lynette Tippett. 2013. Just how expert are "expert" video-game players? Assessing the experience and expertise of video-game players across "action" video-game genres. *Frontiers in Psychology* 4, (2013), 941.

[41] Tessa Lau, Lawrence Bergman, Vittorio Castelli, and Daniel Oblinger. 2004. Sheepdog: learning procedures for technical support. In *Proceedings of the 9th international conference on Intelligent user interfaces* (IUI '04), Association for Computing Machinery, New York, NY, USA, 109–116.

[42] Conor Linehan, George Bellord, Ben Kirman, Zachary H. Morford, and Bryan Roche. 2014. Learning curves: analysing pace and challenge in four successful puzzle games. In *Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play* (CHI PLAY '14), Association for Computing Machinery, New York, NY, USA, 181–190.

[43] Thomas W. Malone. 1982. Heuristics for designing enjoyable user interfaces: Lessons from computer games. In *Proceedings of the 1982 Conference on Human Factors in Computing Systems* (CHI '82), Association for Computing Machinery, New York, NY, USA, 63–68.

[44] Joanna McGrenere and Gale Moore. 2000. Are we all in the same" bloat"? In *Graphics interface*, 187–196.

[45] Edward F. Melcer and Marjorie Ann M. Cuerdo. 2020. Death and Rebirth in Platformer Games. In *Game User Experience And Player-Centered Design*. Springer, Cham, 265–293.

[46] Christie D. Michelsen, Wayne D. Dominick, and Joseph E. Urban. 1980. A methodology for the objective evaluation of the user/system interfaces of the MADAM system using software engineering principles. In *Proceedings of the 18th annual Southeast regional conference* (ACM-SE 18), Association for Computing Machinery, New York, NY, USA, 103–109.

[47] Jonas Molin. 2010. GAID : a practical model of game approachability testing of computer games. diva-portal.org.

[48] Jonas Molin and Charlotte Wiberg. 2014. Game Approachability: Remodelling testing of computer Games with the GAID model. (2014). Retrieved from https://www.academia.edu/download/48952710/04.JAJCS10085.pdf

[49] Jakob Nielsen. 1994. *Usability Engineering*. Morgan Kaufmann.

[50] Janne Paavilainen. 2020. Defining playability of games: functionality, usability, and gameplay. In *Proceedings of the 23rd International Conference on Academic Mindtrek* (AcademicMindtrek '20), Association for Computing Machinery, New York, NY, USA, 55–64.

[51] Suporn Pongnumkul, Mira Dontcheva, Wilmot Li, Jue Wang, Lubomir Bourdev, Shai Avidan, and Michael F. Cohen. 2011. Pause-and-play: automatically linking screencast video tutorials with applications. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (UIST '11), Association for Computing Machinery, New York, NY, USA, 135–144.

[52] John R. Porter and Julie A. Kientz. 2013. An empirical study of issues and barriers to mainstream video game accessibility. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility* (ASSETS '13), Association for Computing Machinery, New York, NY, USA, 1–8.

[53] Jon Radoff. 2011. Game player motivations. *Online:. Data dostepu* 30, (2011).

[54] Sheri Graner Ray. 2010. Tutorials: Learning to play. *Gamasutra*, http://www.gamasutra. com/view/feature/134531/ tutorials_learning_to_play. php (2010).

[55] John Rieman. 1996. A field study of exploratory learning strategies. *ACM Transactions on Computer-Human Interaction* 3, 3 (September 1996), 189–218.

[56] Yvonne Rogers, Helen Sharp, and Jenny Preece. 2011. *Interaction Design: Beyond Human - Computer Interaction*. John Wiley & Sons.

[57] Sebastian Röhlcke, Christian Bäcklund, Daniel Eriksson Sörman, and Bert Jonsson. 2018. Time on task matters most in video game expertise. *PLoS One* 13, 10 (October 2018), e0206555.

[58] S. A. Sabab, A. Khan, P. K. Chilana, J. McGrenere, and A. Bunt. 2020. An Automated Approach to Assessing an Application Tutorial's Difficulty. In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 1–10.

[59] Kathy Sanford, Lisa J. Starr, Liz Merkel, and Sarah Bonsor Kurki. 2015. Serious games: video games for good? *E-Learning and Digital Media* 12, 1 (January 2015), 90–106.

[60] Jason Schreier. *Press Reset: Ruin and Recovery in the Video Game Industry*.

[61] David Williamson Shaffer, Kurt R. Squire, Richard Halverson, and James P. Gee. 2005. Video Games and the Future of Learning. *Phi Delta Kappan* 87, 2 (October 2005), 105–111.

[62] Evan Skolnick. 2014. *Video game storytelling: What every developer needs to know about narrative techniques*. Watson-Guptill Publications.

[63] Lucy A. Suchman and Randall H. Trigg. 1992. Understanding practice: video as a medium for reflection and design. In *Design at work: cooperative design of computer systems*. L. Erlbaum Associates Inc., USA, 65–90.

[64] Katie Salen Tekinbas and Eric Zimmerman. 2003. *Rules of Play: Game Design Fundamentals*. MIT Press.

[65] Gustavo F. Tondello, Rina R. Wehbe, Lisa Diamond, Marc Busch, Andrzej Marczewski, and Lennart E. Nacke. 2016. The Gamification User Types Hexad Scale. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play* (CHI PLAY '16), Association for Computing Machinery, New York, NY, USA, 229–243.

[66] Z Toups, Nicolas Lalone, Sultan A. Alharthi, Hitesh Nidhi Sharma, and Andrew M. Webb. 2019. Making Maps Available for Play: Analyzing the Design of Game Cartography Interfaces. *ACM Transactions on Computer-Human Interactions*. 26, 5 (July 2019), 1–43.

[67] Z Toups, Igor Dolgov, and Elizabeth M. Bonsignore. 2014. A theory of game mechanic signaling for interface design. In *Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play* (CHI PLAY '14), Association for Computing Machinery, New York, NY, USA, 445–446.

[68] Matthew M. (matthew Martin) White. 2012. New tutorials for digital games: game design meets instructional design. doctoral. Memorial University of Newfoundland.

[69] Jason Wuertz, Sultan A. Alharthi, William A. Hamilton, Scott Bateman, Carl Gutwin, Anthony Tang, Z Toups, and Jessica Hammer. 2018. A Design Framework for Awareness Cues in Distributed Multiplayer Games. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–14.