Exploring Help Facilities in Game-Making Software

Dominic Kao
Purdue University
West Lafayette, Indiana
kaod@purdue.edu

ABSTRACT

Help facilities have been crucial in helping users learn about software for decades. But despite widespread prevalence of game engines and game editors that ship with many of today's most popular games, there is a lack of empirical evidence on how help facilities impact game-making. For instance, certain types of help facilities may help users more than others. To better understand help facilities, we created game-making software that allowed us to systematically vary the type of help available. We then ran a study of 1646 participants that compared six help facility conditions: 1) Text Help, 2) Interactive Help, 3) Intelligent Agent Help, 4) Video Help, 5) All Help, and 6) No Help. Each participant created their own first-person shooter game level using our game-making software with a randomly assigned help facility condition. Results indicate that Interactive Help has a greater positive impact on time spent, controls learnability, learning motivation, total editor activity, and game level quality. Video Help is a close second across these same measures.

CCS CONCEPTS

• **Applied computing** → **Computer games**; • **Human-centered computing** → **Empirical studies in HCI**.

KEYWORDS

game making, tutorials, help facilities, text documentation, interactive tutorial, intelligent agent, video

ACM Reference Format:
Dominic Kao. 2020. Exploring Help Facilities in Game-Making Software. In *International Conference on the Foundations of Digital Games (FDG '20), September 15–18, 2020, Bugibba, Malta.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Many successful video games, such as *Dota 2* and *League of Legends* (from *WarCraft 3*), *Counter-Strike* (from *Half-Life*), and the recent *Dota Auto Chess* (from *Dota 2*), are modifications of popular games using game-making or level-editing software. The popular game engine Unity powers 50% of mobile games, and 60% of all virtual reality and augmented reality content [116]. Despite the reach and impact of game-making, very few empirical studies have been done on help facilities in game-making software. For example, in our systematic review of 85 game-making software, we find that the majority of game-making software incorporates text help, while about half contain video help, and only a small number contain interactive help. Given the large discrepancies in help facility implementation across different game-making software, it becomes important to question if different help facilities make a difference in user experience, behavior, and the game produced.

Help facilities can teach users how to use game-making software, leading to increased quality in created games. Through fostering knowledge about game-making, help facilities can better help novice game-makers transition to becoming professionals. While studies on game-making and help facilities do not currently exist, there is good motivation for this topic from gaming. A key study by Andersen et al. [3] suggests that help facilities can be beneficial in complex games (increasing play time by as much as 29%), but their effects were non-significant in simpler games where mechanics can be discovered through experimentation. Because game-making software often presents users with a larger number and higher complexity of choices compared to games [43], game-making is likely a domain in which help facilities play an important role.

In this paper, we start by first reviewing the help facilities in popular game-making software, including game engines and game editors. This allowed us to understand which types of help facilities are present in game-making software, as well as how those help facilities are implemented. This review directly influenced the design of our help facility conditions in our main study. We then describe our game-making software, *GameWorld*, which allows users to create their own first-person shooter (FPS) games. Lastly, we describe a between-subjects experiment conducted on Amazon Mechanical Turk that varied the help facility available to the user. This allowed us to isolate the impact of help facility type while keeping all other aspects of the game-making software identical. In this experiment, we had 5 research questions:

RQ1: Do help facilities lead to higher motivated behavior (time spent, etc.)?
RQ2: Do help facilities improve learnability of controls?
RQ3: Do help facilities improve learning motivation?
RQ4: Do help facilities improve cognitive load?
RQ5: Do help facilities improve created game levels?
RQ6: Does time spent on help facilities vary?

Results show that the interactive help has a substantial positive impact on time spent, controls learnability, learning motivation, cognitive load, game-making actions, and final game level quality. The video help has a similarly positive impact on time spent, learning motivation, cognitive load, and final game level quality.

On the other hand, results show that having no help facility results in the least amount of time spent, lowest controls learnability,

FDG '20, September 15–18, 2020, Bugibba, Malta

ACM ISBN 978-1-4503-8807-8/20/09…$15.00
https://doi.org/10.1145/1122445.1122456

lowest learning motivation, highest cognitive load, lowest game-making actions, and lowest final game level quality. We found that the other help facility conditions (text, intelligent agent, all) generally did not significantly differ from no help, except in cognitive load (text is better than no help, but worse than all other conditions). Finally, we conclude with a discussion of design implications based on the results of the study.

2 RELATED WORK

HCI and games researchers have long been interested in games and learning [40–42, 48, 50, 56, 60, 62, 103, 118, 120]. This includes studies on tutorials [3], differences in frequent versus infrequent gamers' reactions to tutorials [82], leveraging reward systems [32], encouraging a growth mindset, or the idea that intelligence is malleable [55, 86, 87], avatars [57, 59, 63, 66], embellishment [58, 65], and many more. AI and games researchers have also begun to take interest, such as the automatic generation of video game tutorials [36, 37], and the adaptation of tutorials to individual user skill levels [8].

In this section, we begin with an overview of software learnability and multimedia learning. We then review the types of help facilities that are found most often in game-making software. These are text documentation, video tutorials, and interactive tutorials. We also investigate intelligent agents. Despite not being present in most game-making software, intelligent agents have been widely explored as an effective means of teaching in the academic literature. Although there are many other types of help (e.g., showing a random tip on startup) and variations thereof [3], our focus is on: 1) Core help facilities commonly available in game-making software, and 2) Common implementations of those facilities. Both this literature, and the review of game-making software, provides a baseline for developing our own help facility conditions.

2.1 Software Learnability

Software learnability is a general term that refers to learning how to use a piece of software. Software learnability can be measured along different dimensions, including task metrics (i.e., task performance), command metrics (i.e., based on commands issued by the user), mental metrics (i.e., related to cognitive processes), and subjective metrics (i.e., learnability questionnaires). In this paper, we triangulate across these multiple categories by leveraging expert game level ratings (task), total game-making actions (command), cognitive load measures (mental), and a questionnaire assessing learnability of controls (subjective), to gauge the effects of help facilities on game-making software learnability. One important aspect of our study is cognitive load—this refers to human working memory usage [112]. Here, we are interested in studying the amount of cognitive load experienced by users in each of the help facility conditions. Although help facilities may help users moderate cognitive load through scaffolding the game-making activity, they may also lead to negative impacts, e.g., overwhelming the user with information [122].

2.2 Multimedia Learning Theory

Multimedia learning theory illustrates the principles which lead to the most effective multimedia (i.e., visual and auditory) teaching materials [77]. These principles include the *multimedia principle* (people learn better from words and pictures than from words alone), the *spatial contiguity principle* (people learn better when corresponding words and pictures are presented near rather than far from each other), and the *temporal contiguity principle* (people learn better when corresponding words and pictures are presented simultaneously rather than successively) [78]. We utilize this framework as one internal guide in developing our help facilities. In the remaining sections, we survey different modalities of help facilities.

2.3 Text-Based Help Facilities

Early forms of computing documentation originated from the advent of commercial mainframe computing [125]. Much of the research on text help is dedicated to improving the user experience of computing documentation. Converging evidence suggests that user frustration with computers is a persistent issue that has not been satisfactorily ameliorated by accompanying documentation [71, 80, 109]. Over the past few decades, researchers have proposed several methods for improving the user experience of computing documentation, including standardizing key software terminology and modes of expression [6, 119]; automatically generating documentation material [93]; using semantic wiki systems to improve and accelerate the process of knowledge retrieval [24]; and drastically shortening text manuals by eliminating large sections of explanation and elaborations [18]. A significant issue in this research, however, is the dearth of systematic reviews and comprehensive models for evaluating the efficacy of text-based help facilities [126]. As a result, it remains difficult to determine both the utility of computing documentation for users and developers and whether the benefits of production outweigh the costs [25].

In one study of tutorials and games, text-based tutorials were associated with a 29% increase in length of play in the most complex game; there was no significant increase with the tutorials for simpler games, which suggests that investing in the creation of tutorials for simpler games may not be worth it [3]. Researchers have stated that official gaming documentation faces a gradual but substantial decline [38]. This can be attributed to several factors. Scholars and consumers of computer games typically agree that the best gaming experience is immersive and, therefore, that necessitating any documentation to understand gameplay is a hindrance; at the same time, complex games that lack text-based help facilities are frequently criticized for having steep learning curves that make immersion difficult [81]. Moreover, researchers have argued that there is a lack of standardization across different games, and that help documentation is often written by game developers themselves (often through simply augmenting internal development texts), which has decreased the efficacy of text-based documentation [2, 38, 81, 121].

2.4 Interactive Tutorial Help Facilities

Since the mid-1980s, early research has sought to understand the effectiveness of interactive tutorials on learning [20, 73, 74]. Interactive tutorials have been found to be especially effective in subjects that benefit from visualizing concepts in detail; engineering students, for example, can interact with graphical representations of objects that are difficult or impossible to do so with still images

[74]. Interactive tutorials have been found to be highly effective in learning problem-solving [29, 114]. Additionally, interactive tutorials have been found to be superior to non-interactive methods in learning factual knowledge [73], database programming [90], medical library instruction [5], and basic research skills [111].

Game designers often emphasize the importance of user experimentation while learning new concepts [97]. This experimentation, James Gee argues, should take place in a safe and simplified environment where mistakes are not punished [34]. Kelleher and Pausch have shown that restricting user freedom improves tutorial performance [68]. Using the Alice programming environment, they find that with an interactive tutorial called Stencils, users are able to complete the tutorial faster and with fewer errors than a paper-based version of the same tutorial. Molnar and Kostkova found that children 10-13 years of age reacted positively to the incorporation of an interactive tutorial that guides the player explicitly through game mechanics [83]. On the other hand, participants that did not play the interactive tutorial found the game more awkward [84]. Frommel et al. found that in a VR game, players taught more interactively had higher positive emotions and higher motivation [30].

2.5 Intelligent Agent Help Facilities

The persona effect was one of the earliest studies that revealed that the mere presence of a life-like character in a learning environment increased positive attitudes [61, 72]. Intelligent agents are on-screen characters that respond to feedback from users in order to enhance their experience [94, 123]. These are often used to effectively tailor learning environments for individual students [94, 123]. Intelligent agents can help to personalize learning more effectively than standard teaching tools [94], and allow for human-like interaction between the software and the user that would not otherwise be possible [10, 108]. Intelligent agents have been integrated into several games whose purpose is to teach the player. The TARDIS framework uses intelligent agents in a serious game for social coaching for job interviews [4]. Other educational games have utilized intelligent agents to teach the player number factorization [22, 23], the Java compilation process [35], and computational algorithms [31].

2.6 Video Tutorial Help Facilities

Video-based tutorials utilize the modalities of audio, animation, and alphabetic text. Research has shown that user performance is increased when animation is combined with an additional semiotic mode, such as sound or words [79]. Video animation is effective for recall when illustrating highly visual facts, concepts, or principles [99] (p.116). For instance, video tutorials can display a task sequence in the same way a user would see it on their own computer screen, leading to congruence between the video and the real-life task execution [115]. Many studies have shown that video tutorials can be highly effective [14, 117, 124]. For example, one study found that 24.2% of students without videos failed a course on introductory financial accounting, whereas the failure rate was only 6.8% among students that had the videos available [14]. Another study that compared text tutorials to video tutorials for learning software tools found that both types of tutorials had their advantages. Namely, video tutorials were preferred for learning new content; however, text tutorials were useful for looking up specific information [54]. Video walkthroughs are common instructional tutorials used in games to help players overcome a game's challenges through the imitation of actions [12, 17, 85]. For example, a classroom study supplemented the use of video games with walkthroughs, and found that students found the video-based walkthroughs more helpful than the text-based ones [12].

2.7 Game-Making

Academic interest in game-making has its roots in constructionism: the theory of learning in which learners construct mental models for understanding the world [92]. Early manifestations included "Turtle Geometry," an environment for programming an icon of a turtle trailing lines across a computer display. Research at the intersection of HCI, game-making, and education has shown that game-making has promise for increasing engagement, knowledge, and skills in a variety of domains [1, 26, 33, 44, 45, 52, 53, 64, 100–102]. However, despite an extensive literature on game-making and education, game-making *software* is seldom studied. [75] is one rare example in which 8 game-making tools were contrasted on their immersive features. Given the scarcity of work on game-making software, it is difficult to predict which types of help facilities will be most effective. Even in games, despite employing a wide variety of tutorial styles, the relative effectiveness of these styles is not well understood [3]. The main goal of the current study is to explore the effects of help facilities within game-making software.

3 INTERSECTIONALITY BETWEEN PLAY AND MAKING

Before proceeding, it is crucial to discuss our approach in studying game-making software. Frequently, in developing this work and discussing it with others, we often broached the topic of what *making* is, and what *play* is. Yet in trying to define these terms, even in a specific context such as games, we reach a deep philosophical impasse. Huizinga is well-known to be the progenitor of one of the most widely used (but widely contested) definitions of *play* [46, 107]. Piaget made the following observation: "the many theories of play expounded in the past are clear proof that the phenomenon is difficult to understand" [95]. Instead of attempting to delineate the two terms, we argue that it is precisely their intersectionality that needs further theoretical and empirical grounding. We argue that strictly categorizing an activity as *play* or *making* threatens to constrain researchers to drawing on traditional epistemologies inherent to how the terms have been defined previously, rather than building new interdisciplinary bridges which shed light on both parallels and divergences.

For example, we find in the next section that game-making software often appears to fall on a continuum that is neither fully software nor fully game. We could argue that LittleBigPlanet should be categorized as a game, and that Unity should be categorized as software. Yet elements of play and making are present even in these more extreme examples—in Unity, users engage in frequent playtesting, in part to see if their created game is "fun". Therefore, there appear to be a number of both parallels and divergences between play and making, and their degree of overlap in any given context

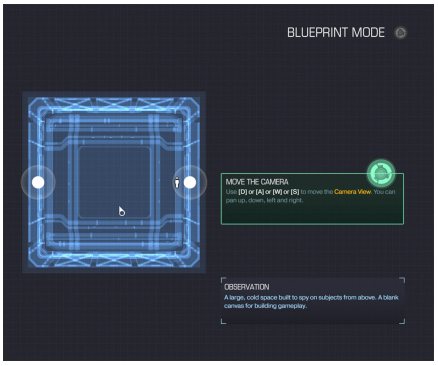

Figure 1: Doom's SnapMap interactive tutorial.

Figure 2: LittleBigPlanet 3 interactive tutorial.

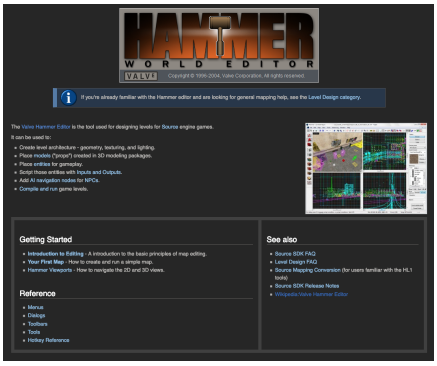

Figure 3: Valve's Hammer editor text doc.

will inevitably depend on the definitions that one has decided to apply. In this paper, we avoid strict categorization of game-making software as being a pure "game" or pure "software"—this allows our survey to more flexibly encompass a wide range of game-making systems, regardless of whether they exist as independent environments or embedded inside the ecology of a game.

4 REVIEW OF GAME-MAKING SOFTWARE

Before designing our experiment, we reviewed 85 different game-making software. This includes both game engines and official level editors. We retrieved the list of software based on commercial success and popularity (Wikipedia/Google), critical reception (Metacritic), and user reception (Slant.co). For example, Slant.co shows user-driven rankings for "What are the best 2D game engines?" and "What are the best 3D game engines?".

Each piece of software was first installed on an appropriate device, then explored by 2 experienced (8+ years of professional game development experience) game developers independently for 1 hour. Each individual then provided their own summary of the software and the help facilities available. At this stage, all possible help facilities were included, such as interactive tutorials, startup tips, community forums, and so on. In some rare instances, we excluded software prior to review that did not come from an official source. (One notable example is Grand Theft Auto V, which does not have official modding tools.) For examples, see Figure 1, 2, and 3.

Next, we condensed our review into a table summarizing the different types of help available for each software. The table was coded independently by the 2 developers, then discussed and re-coded repeatedly until consensus was reached. At this stage, we made the distinction between help facilities contained directly in the software versus external help facilities. External help facilities included online courses, community forums, and e-mailing support. These types of help fall outside the main intent of our current research, which is to study help facilities contained in the software itself and were therefore excluded. An exception was made for common internal help facilities that were external, so long as they came from an official source and so long as a link was included to those help facilities directly from within the software (e.g., online text documentation, videos, etc.). Unofficial sources of help were not included. Finally, types of help that were contained within the software but were not substantive enough to warrant their inclusion as a core help facility (such as a random tip appearing each time the software boots) were excluded.

	Text Documentation	Interactive Tutorial	Intelligent Agent	Video Tutorial
Unity Engine	✓	✓	✗	✓
Unreal Engine	✓	✓	✗	✓
GameMaker Studio 2	✓	✗	✗	✓
Godot Engine 3	✓	✗	✗	✓
CryEngine	✓	✗	✗	✓
Cocos2d-x	✓	✗	✗	✓
Buildbox	✓	✗	✗	✓
StarCraft	✓	✗	✗	✗
StarCraft 2	✓	✗	✗	✗
Legend of Grimrock 2	✓	✗	✗	✓
Neverwinter Nights	✓	~	✗	✗
Neverwinter Nights 2	✓	✗	✗	✗
Doom (2016)	✓	✓	✗	✗
Dota 2	✓	✗	✗	✗
Half-Life	✓	✗	✗	✗
Half-Life 2	✓	✗	✗	✗
Garry's Mod	✗	~	✗	✗
Shadowrun Returns	✓	✗	✗	✗
Tenchu 2	✗	✗	✗	✗
Construct 2	✓	✗	✗	✓
RPG Maker MV	✓	✓	~	✗
WarCraft 2	✓	✗	✗	✗
WarCraft 3	✓	✗	✗	✗
LittleBigPlanet	✗	✓	~	✓
LittleBigPlanet 2	✗	✓	~	✓
LittleBigPlanet 3	✗	✓	~	✓
Torchlight	✗	✗	✗	✓
Torchlight 2	✓	✗	✗	✓
Skyrim	✓	✗	✗	✗
Project Spark	✓	✓	✗	✓
Minecraft	✓	✓	✗	✓
Morrowind	✓	✗	✗	✗
Halo 5: Guardians	✓	✗	✗	✗
Super Mario Maker	✓	✓	~	✗
TrackMania	✓	✓	✗	✓
ShootMania	✓	✓	✗	✓
Dying Light	✓	✗	✗	✓
Portal 2	✓	✓	✗	✗
Duke Nukem 3D	✗	✗	✗	✗
Left 4 Dead	✓	✗	✗	✗
Left 4 Dead 2	✓	✗	✗	✗
AppGameKit	✓	✗	✗	✓
MonoGame	✓	✗	✗	✓

	Text Documentation	Interactive Tutorial	Intelligent Agent	Video Tutorial
Amazon Lumberyard	✓	✓	✗	✓
Shiva Engine	✓	✗	✗	✓
Hero Engine	✓	✗	✗	✓
ImpactJS	✓	✗	✗	✓
Turbulenz	✓	✗	✗	✗
JMonkeyEngine	✓	✗	✗	✓
Torque 3D	✓	✗	✗	✗
Panda 3D	✓	✗	✗	✗
Corona	✓	✗	✗	✓
Unigine	✓	✗	✗	✓
Leadwerks	✓	✗	✗	✓
Wintermute Engine	✓	✗	✗	✗
ORX Engine	✓	✗	✗	✗
libGDX	✓	✗	✗	✗
Urho 3D	✓	✗	✗	✗
GameSalad	✓	✗	✗	✓
ClickTeam Fusion	✓	✗	✗	✓
Stencyl	✓	✗	✗	✓
GameGuru	✓	✗	✗	✓
Axis Game Factory	✓	✓	✗	✓
CopperCube	✓	✗	✗	✓
Phaser	✓	✗	✗	✓
Xcode	✓	✗	✗	✓
Android Studio	✓	✓	✗	✓
PlayCanvas	✓	✗	✗	✓
GamePlay	✓	✗	✗	✗
ZGameEditor	✓	✗	✗	✓
Gamebryo	✗	✗	✗	✗
Polycode	✗	✗	✗	✗
Spring Engine	✓	✗	✗	✗
Vanda	✓	✗	✗	✓
Angel2D	✓	✗	✗	✗
Gideros	✓	✗	✗	✓
LE 2D	✓	✗	✗	✓
GDevelop	✓	✗	✗	✓
Pygame	✓	✗	✗	✗
Allegro	✓	✗	✗	✗
HaxePunk	✓	✗	✗	✗
HaxeFlixel 2D	✓	✓	✗	✗
Monkey 2	✓	✗	✗	✗
Flixel	✓	✗	✗	✗
Babylon.js	✓	✗	✗	✓

Figure 4: Game-making software and their official help facilities. Green means "Yes" (✓), orange means "Somewhat" (~), and red means "No" (×).

Our final table contained the following categories of help facilities: text documentation, interactive tutorial, and video tutorial. In addition, intelligent agent was included as a result of our earlier rationale. See Figure 4. Overall, the review shows that text documentation is prevalent in the majority of game-making software (89.4%). Official video tutorials are present in approximately half of game-making software (52.9%). A lesser number of game-making software contain interactive tutorials (20.0%). Finally, no games contained an intelligent agent that responded to user choices (0.0%). A few game-making software contained a character that would lead the player through a tutorial, but these were purely aesthetic and were not full-fledged intelligent agents.

5 THE GAME-MAKING SOFTWARE

We developed a game-making software called *GameWorld*[1]. *GameWorld* was developed using a spiral HCI approach by repeatedly designing, implementing, and evaluating prototypes in increasingly complex iterations. Evaluation of prototypes was performed with experienced game developers known to the author. *GameWorld* was developed specifically for novice game-makers, and allows users to create a first-person shooter game without any coding.

Figure 5 shows the main interface elements. The top of the interface is primarily dedicated to object-related actions. The left side allows additional object manipulations. For example, objects in *GameWorld* are typically aligned to an underlying grid. However, the user can hold down Control while modifying position, rotation, or scale, which ignores the grid alignment and gives the user more flexibility. Therefore, the "Align" buttons allow for objects to be snapped back into grid alignment. Objects can also be grouped (for easier management), and be made dynamic (which means they are moveable during play, for instance from collisions with bullets or player models). Dynamic is an important modifier for certain objects, such as a door, which consists of a door frame, a door joint, and a door which has the dynamic modifier enabled.

Objects. There are 36 pre-made objects that users can place. These include simple objects (e.g., a sphere), to more complex objects (e.g., a guard room). Players can also create their own objects, for example by grouping objects together and saving them as a "pre-fab". Objects can be textured and colored. There are 76 pre-made textures that users can choose. There are 146 color choices.

Special Objects. Special objects are non-standard objects like door joints, invisible walls, lights, player and enemy spawn points, and trees. These are manipulated in the same way as normal objects.

Level Properties. Within the level properties menu, players can modify the starting health of the player, number of enemies, whether enemies should respawn after death (and how often), certain modes useful for testing (e.g., player invincibility), etc. Some of these settings can also be changed during play testing in the pause menu.

Builder Tool. The builder tool allows users to create arbitrary objects using cubes, each cube corresponding to one grid volume.

6 DEVELOPING HELP FACILITIES

In developing the help facility conditions, our primary objectives were: 1) Consistent quality across the different help facilities, and 2) Realistic implementations similar to current game-making software. To this end, we sought freelancer game developers to help with "Providing Feedback on Game-Making Software". We told game developers that we wanted critical feedback on game-making software being developed. We hired a total of 15 professional game developers, each with an average of 4 years (SD=2.0) of game development experience. Each game developer had worked with at least 3 different game engines, with more than half of the developers having experience with 5+. Developers all had work experience and portfolios which reflected recent game development experience (all within one year). These game developers provided input throughout the help facility development process. Game developers provided feedback at three different times during the the creation of our help facility conditions: During the initial design, after initial prototypes, and after completing the polished version. Game developer feedback was utilized to create help facilities that developers thought would be helpful, as well as similar to existing implementations in game-making software. Additionally, the authors of this paper incorporated multimedia learning principles wherever possible in developing the help facilities. Finally, a questionnaire was administered to the developers to verify that our objectives of consistent quality and realistic implementations was satisfied.

6.1 Conditions

We created 6 help facility conditions:

- No Help
- Text Help
- Interactive Help
- Intelligent Agent Help
- Video Help
- All Help

See Figures 6, 7, 8, and 9. The No Help condition was a baseline control condition. The All Help condition contained all of the help facilities. Every help facility contained the same identical quick tutorial which consisted of learning *navigation* (moving around the editor), *creating the world* (creating and rotating objects), *adding enemies* (creating enemy respawn points), *level configuration* (changing additional level parameters), and *play* (play testing). Upon completing the quick tutorial, users will have learned all of the concepts necessary to create their own level. Help facilities are integrated directly into the application to facilitate data tracking.

When the editor first loads, the user is presented with the dialog "Go to X now?" (X is replaced by Text Help, Interactive Help, Intelligent Agent Help, or Video Help). If the user clicks "Yes", the help facility is opened. If the user presses "No" then the user is notified that they can access the help facility at any time by clicking on the help icon. This happens only once for each of the help facility conditions. In the All Help condition, every help facility is presented in the opening dialog in a randomized order (randomized per-user), with each help facility presented as a button and "No Thanks" at the bottom of the list. In the No Help condition, no opening dialog is presented, and pressing the help icon brings up the dialog "Currently Unavailable".

6.1.1 Text Help. The Text Help window is a document that contains a menu bar and links to quickly navigate to different sections of the text help. The Text Help window can be closed or minimized.

[1]Demo: https://youtu.be/O7_VH0IyWdo

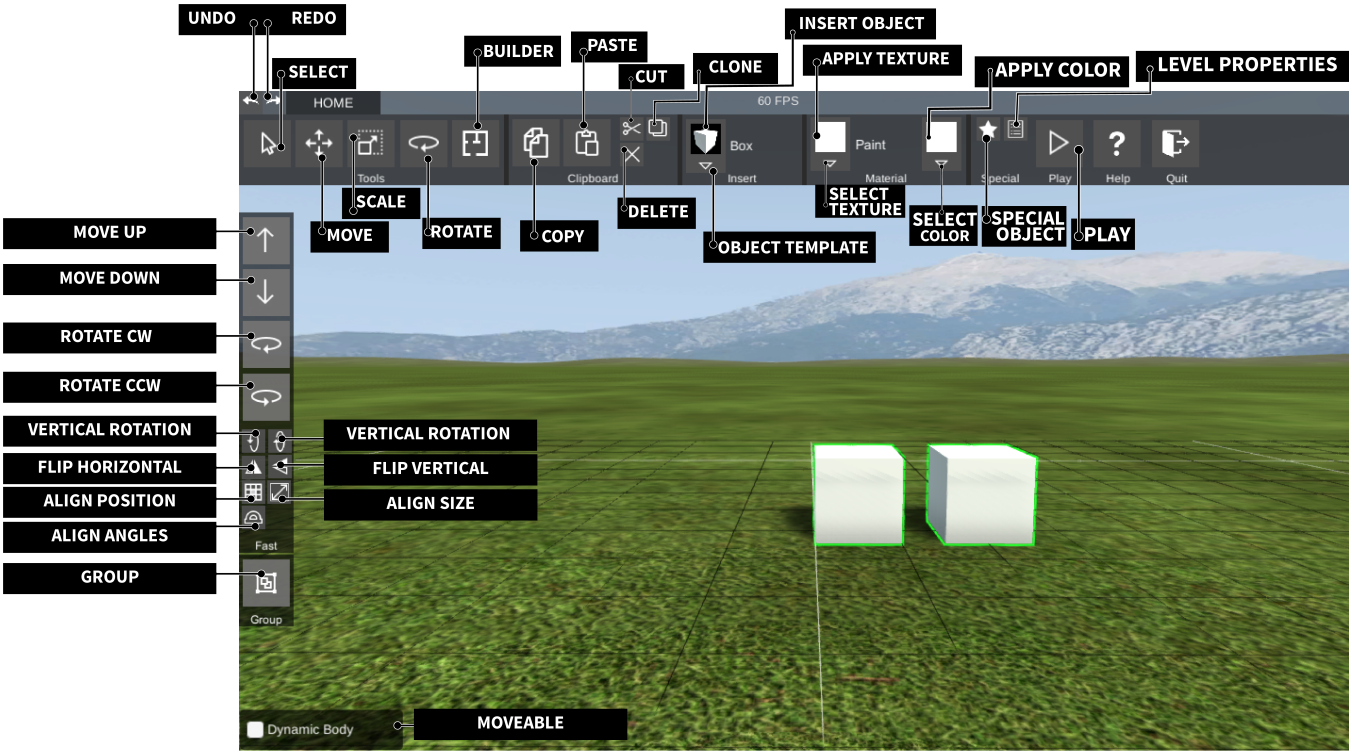

Figure 5: Interface overview. Each interface element has a corresponding tooltip.

In either case, the Text Help window will re-open at the same location the user was at previously. When the Text Help is minimized, it appears as an icon near the bottom of the editor screen with the caption "Text Help".

The Text Help contains the quick tutorial. However, after the quick tutorial, the Text Help contains additional reference material. This includes in-depth (advanced) documentation on navigation, level management, objects management, special objects, and play testing. Additional information is provided that is not covered in the quick tutorial (e.g., hold down shift to amplify navigation, how to peek from behind corners during play, how to add lighting, etc.). Screenshots are provided throughout to add clarity.

6.1.2 Interactive Help. The Interactive Help provides the quick tutorial interactively. Players are limited to performing a specific action at each step (a darkened overlay only registers clicks within a cut-out area). When users are presented with information that does not require a specific action, users can immediately click "Next". Users can close the interactive tutorial at any time. If a user re-opens a previously closed interactive tutorial, the tutorial starts at the beginning—this behavior is consistent with existing interactive tutorials in game-making software.

6.1.3 Intelligent Agent Help. The Intelligent Agent Help is an intelligent agent that speaks to the user through dialog lines. A female voice actor provided the dialog lines of the intelligent agent. The intelligent agent has gestures, facial expressions, and lip movements that are synchronized to the audio. This was facilitated using the *SALSA With RandomEyes* and *Amplitude for WebGL* Unity packages. For gestures, we created custom talk and idle animations.

When the Intelligent Agent Help is activated, it provides several options: 1) Quick Tutorial (identical to the interactive tutorial and everything is spoken by the intelligent agent), 2) Interface Questions (clicking anywhere on the screen provides an explanation—this also works with dialogs that are open such as level properties), 3) Other Questions (a pre-populated list of questions weighted by the user's least taken actions, e.g., if the user has already accessed level properties, this particular question will appear on a later page; there are three pages of common questions, e.g., "How do I add enemies?"). The agent can be closed and re-activated at any time.

6.1.4 Video Help. The Video Help provides the quick tutorial in a video format (4 minutes, 27 seconds). Audio is voiced by the same female actor as for the Intelligent Agent Help. Captions are provided at the beginning of each section (e.g., "Navigating"). A scrubber at the bottom allows the user to navigate the video freely. The Video Help can be closed or minimized. In either case, the Video Help window will re-open at the same location the user was at previously. When the Video Help is minimized, it appears as an icon near the bottom of the editor screen with the caption "Video Help".

6.1.5 All Help. In the All Help condition, users have access to all help facilities. Help facilities work the same as described, except only one help facility can be active at a time.

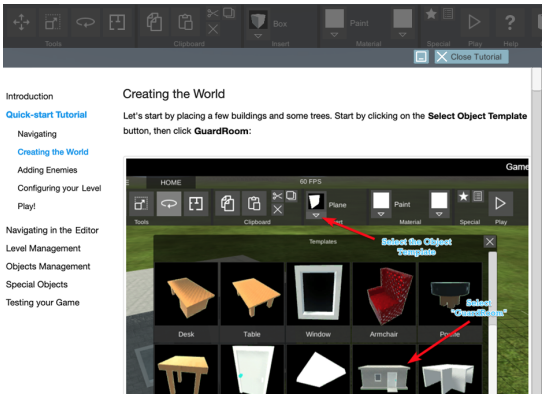

Figure 6: Text Help condition.

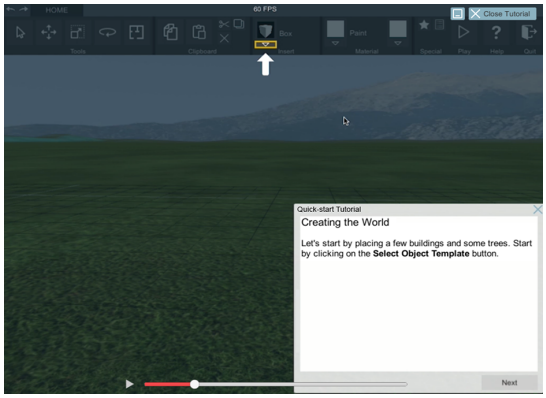

Figure 7: Video Help condition.

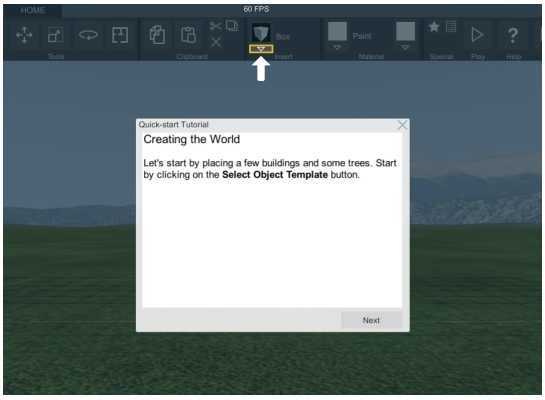

Figure 8: Interactive Help condition.

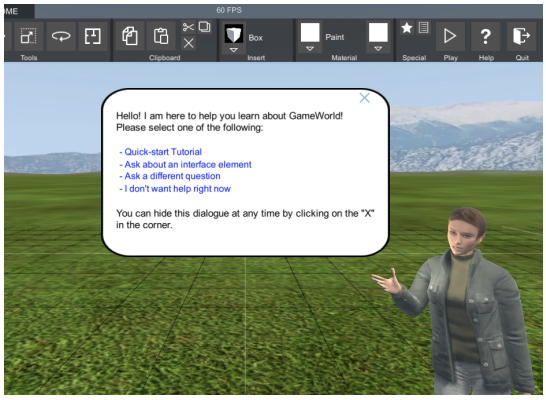

Figure 9: Intelligent Agent Help condition.

6.2 Validating Help Facilities

The feedback of the professional game developers played an important role in the creation of our help facilities. For example, the initial prototype of the Text Help was a simple text document with images. However, developers commented that most game-making software would contain easy-to-navigate documentation. Therefore, we enhanced the Text Help with a menu bar that contained section links.

After completing the final versions of the help facilities, we asked the game developers to answer a short survey. Each game developer, on their own, explored each help facility in a randomized order for at least 30 minutes. After each help facility, game developers anonymously answered two questions: "Overall, I felt that the quality of the X was excellent," and "Overall, I felt that the X was similar to how I would expect it to be implemented in other game-making software," on a scale of 1:*Strongly Disagree* to 7:*Strongly Agree*.

A one-way ANOVA found no significant effect of help facility condition on game developer quality ratings at the $p<.05$ level [$F(3,56) = 0.24$, $p = 0.87$]. The average quality score for each help facility was M=6.0, SD=1.3 (Interactive Tutorial), M=5.9, SD=1.1 (Video Tutorial), M=6.1, SD=0.8 (Text Tutorial), M=5.9, SD=0.8 (Intelligent Agent). A one-way ANOVA found no significant effect of help facility condition on game developer similar implementation ratings at the $p<.05$ level [$F(3,56) = 0.35$, $p = 0.79$]. The average similarity scores for each help facility was M=6.3, SD=0.7 (Interactive Tutorial), M=6.1, SD=0.7 (Video Tutorial), M=6.3, SD=0.8 (Text Tutorial), M=6.2, SD=0.7 (Intelligent Agent).

6.3 Validating Frame Rate

To ensure the validity of the experiment, one of our initial goals was to normalize frames per second across the help facilities. A lower frames-per-second count while one of the help facilities was active would present a possible experiment confound. We wanted to ensure that, in particular, the Intelligent Agent which is a 3D model that moved with gestures and facial expressions in a WebGL application, did not create performance issues and a possible degradation of the experience.

For testing, we used a 2018 PC (Windows 10) and a 2012 Macbook Pro (MacOS High Sierra). The PC had an Intel Core i7-7700k CPU (4.20 GHz), an NVIDIA GeForce GTX 1070, and 16 GB of RAM. The Mac had an Intel Core i5 (2.5 GHz), an Intel HD Graphics 4000 GPU, and 6 GB of RAM. Both systems used Firefox Quantum 63.0 to run the Unity WebGL game and for performance profiling.

We produced a 1-minute performance profile for each machine and for each help facility. In the case of Text Help, Interactive Help, and Intelligent Agent Help, interactions occurred at a reading speed of 200 words per minute [127]. We produced a performance profile for each machine and for each help facility. All help facilities

were within ~1 fps: intelligent agent (PC: 59.14 fps, Mac: 59.04), interactive tutorial (PC: 60.00, Mac: 59.11), text tutorial (PC: 60.00, Mac: 59.43), video tutorial (PC: 60.00, Mac: 58.74).

7 METHODS

7.1 Quantitative Measures

7.1.1 Learnability of Controls. The "Controls" subscale from the Player Experience of Need Satisfaction (PENS) scale [106] was adapted for use in this study. This consisted of 3 questionnaire items as follows: "Learning GameWorld's controls was easy", "GameWorld's controls are intuitive", and "When I wanted to do something in GameWorld, it was easy to remember the corresponding control". Cronbach's alpha was 0.86.

7.1.2 Learning Motivation Scale. Learning motivation was captured using a scale adapted from [47] which consisted of 7 items on a 6-point Likert scale (1: *Strongly Disagree* to 6: *Strongly Agree*), e.g., "I would like to learn more about GameWorld". Cronbach's alpha was 0.93.

7.1.3 Cognitive Load. Cognitive load used measures adapted from [88] and [113]. It consists of 8 items on a 6-point Likert scale (1: *Strongly Disagree* to 6: *Strongly Agree*). There are two sub-scales: mental load (e.g., "GameWorld was difficult to learn for me"), and mental effort (e.g., "Learning how to use GameWorld took a lot of mental effort"). Cronbach's alpha was 0.90 and 0.85.

7.1.4 Game Quality Ratings. Users were asked to rate their final game level on the dimensions of: "Aesthetic" (Is it visually appealing?), "Originality" (Is it creative?), "Fun" (Is it fun to play?), "Difficulty" (Is it difficult to play?), and "Overall" (Is it excellent overall?) on a scale of 1: *Strongly Disagree* to 7: *Strongly Agree*.

Expert ratings were given by 3 QA testers we hired. All QA testers had extensive games QA experience. The 3 QA testers first underwent one-on-one training with a GameWorld expert for one hour. QA testers then reviewed 250 game levels on their own without scoring them. QA testers were then given 50 game levels at random to rate. The GameWorld expert provided feedback on the ratings, and the game levels were rescored as necessary. Afterwards, QA testers worked entirely independently.

All 3 QA testers were blind to the experiment—the only information they received was a spreadsheet containing links to each participant's game level. Each game level was played by the QA tester before being rated. They were debriefed on the purpose of their work after they completed all 1646 ratings. The 3 QA testers each spent an average of 64 hours (SD=9.3) over 3 weeks, at $10 USD/hr.

7.1.5 Total Time. We measure both total time, and time spent in each help facility. For all types of help, this is the amount of time that the help is on-screen (and maximized if it is Text Help or Video Help).

7.1.6 Other Measures. We were additionally interested in whether the player activated the help immediately on startup and how many total game-making actions were performed (this was an aggregate measure that combined object creations, object manipulations, etc.).

7.2 Participants

After a screening process that disqualified participants with multiple surveys with zero variance, multiple surveys with ±3SD, or a failed attention check, 1646 Amazon Mechanical Turk participants were retained. The data set consisted of 976 male, and 670 female participants. Participants were between the ages of 18 and 73 (M = 32.3, SD = 9.6), were all from the United States, and could all read/write English.

7.3 Design

A between-subjects design was used: help facility condition was the between-subject factor. Participants were randomly assigned to a condition.

7.4 Protocol

Participants filled out a pre-survey assessing previous experience playing games, programming, and creating games (conditions did not differ significantly across any of these measures, p=0.320, p=0.676, p=0.532). Then for a minimum of 10 minutes, each participant interacted with *GameWorld*. After the 10 minutes had passed, the quit button became active and participants could exit at any time. After quitting, participants completed the PENS, the learning motivation scale, and the cognitive load scale. Participants then provided ratings on their game levels before filling out demographics.

7.5 Analysis

Separate MANOVAs are run for each separate set of items—*PENS, User Level Ratings, Expert Level Ratings*; with the independent variable—*help facility condition*. To detect the significant differences between badge conditions, we utilized one-way MANOVA. These results are reported as significant when p<0.05 (two-tailed). Prior to running our MANOVAs, we checked both assumption of homogeneity of variance and homogeneity of covariance by the test of Levene's Test of Equality of Error Variances and Box's Test of Equality of Covariance Matrices; and both assumptions were met by the data. For individual measures, we use one-way ANOVA.

8 RESULTS

RQ1: Do help facilities lead to higher motivated behavior?

The Interactive Help and Video Help promoted greater time spent. The Interactive Help promoted a higher number of actions. The No Help condition results in the least time spent and the lowest number of actions.

A one-way ANOVA found a significant effect of help facility condition on time spent at the p<.05 level [$F(5,1640) = 10.23$, $p < 0.001$, $\eta_p^2 = 0.03$]. Post-hoc testing using Tukey HSD found that participants in both the Interactive Help and Video Help conditions spent a longer total time than participants in any of the four other conditions, p<.05, d in the range of 0.27–0.46. See Figure 10.

A one-way ANOVA found a significant effect of help facility condition on total game-making actions at the p<.05 level [$F(5,1640) = 4.22$, $p < 0.001$, $\eta_p^2 = 0.01$]. Post-hoc testing using Tukey HSD found that participants in the Interactive Help condition performed a higher number of actions than Text Help (d=0.23), Intelligent Agent Help (d=0.21), All Help (d=0.22), and No Help (d=0.33), p<.05. See Figure 11.

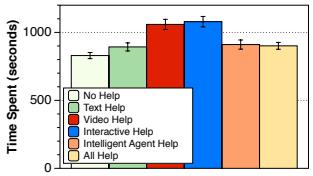

Figure 10: Time spent (+/- SEM).

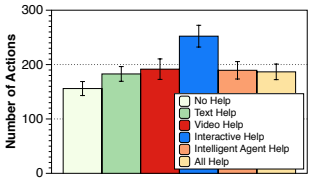

Figure 11: Editor actions (+/- SEM).

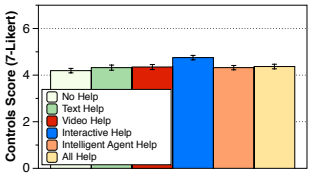

Figure 12: Controls score (+/- SEM).

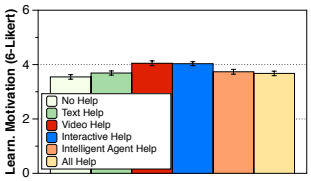

Figure 13: Learning mot. (+/- SEM).

RQ2: Do help facilities improve learnability of controls?

The Interactive Help promoted controls learnability.

A one-way ANOVA found a significant effect of help facility condition on the PENS controls score at the p<.05 level [F(5,1640) = 3.96, p < 0.005, η_p^2 = 0.01]. Post-hoc testing using Tukey HSD found that participants in the Interactive Help condition had a higher PENS controls score than participants in any of the other conditions except All Help and Video Help, p <.05, *d* in the range of 0.27–0.34. See Figure 12.

RQ3: Do help facilities improve learning motivation?

The Interactive Help and Video Help promoted learning motivation. No Help results in the lowest learning motivation.

A one-way ANOVA found a significant effect of help facility condition on learning motivation at the p<.05 level [F(5,1640) = 6.42, p < 0.001, η_p^2 = 0.02]. Post-hoc testing using Tukey HSD found that participants in both the Interactive Help and Video Help conditions had higher learning motivation than participants in any of the other conditions except Intelligent Agent Help, p<.05, *d* in the range of 0.27–0.37. See Figure 13.

RQ4: Do help facilities improve cognitive load?

All conditions had lower cognitive load relative to No Help, except Text Help. Interactive Help, Intelligent Agent Help, and All Help have lower cognitive load than Text Help.

A one-way ANOVA found a significant effect of help facility condition on mental load at the p<.05 level [F(5,1640) = 8.14, p < 0.001, η_p^2 = 0.02]. Post-hoc testing using Tukey HSD found that participants in the No Help condition had a higher mental load than participants in any other condition except Text Help, p<.005, *d* in the range of 0.32–0.39. Participants in the Text Help condition had a higher mental load than Interactive Help (*d*=0.31), Intelligent Agent Help (*d*=0.33), and All Help (*d*=0.28), p <.05.

A one-way ANOVA found a significant effect of help facility condition on mental effort at the p<.05 level [F(5,1640) = 8.29, p < 0.001, η_p^2 = 0.03]. Post-hoc testing using Tukey HSD found that participants in the No Help condition exerted higher mental effort than participants in any other condition, p<.005, *d* in the range of 0.15–0.42. Participants in the Text Help condition exerted higher mental effort than Interactive Help (*d*=0.26) and Intelligent Agent Help (*d*=0.28), p <.05. See Figure 14.

RQ5: Do help facilities improve created game levels?

The Interactive Help and Video Help led to the highest quality game levels, both from the user's perspective and expert ratings. No Help leads to the lowest quality.

The MANOVA was statistically significant across help facility conditions across the self-rated game level quality dimensions, F(25, 6079) = 1.53, p <.05; Wilk's λ = 0.977, η_p^2 = 0.01. ANOVAs found that the effect was significant across all dimensions except difficulty, p <.05, η_p^2 in the range of 0.01–0.02. Posthoc testing using Tukey HSD found that for aesthetic, originality, fun, and overall: Both Interactive Help and Video Help were significantly higher than No Help, p <.05, *d* in the range of 0.25–0.39.

For expert ratings, intraclass correlation across the three raters was ICC=0.83 (two-way random, average measures), indicating high agreement. The MANOVA was statistically significant across help facility conditions across the expert-rated game level quality dimensions F(25, 6079) = 5.97, p <.001; Wilk's λ = 0.914, η_p^2 = 0.02. ANOVAs found that the effect was significant across all dimensions, p <.005, η_p^2 in the range of 0.02–0.06. Posthoc testing using Tukey HSD found that Interactive Help and Video Help were highest across all dimensions (significant values: p <.05, *d* in the range of 0.22–0.75). On the other hand, No Help was lowest across all dimensions (significant values: p <.05, *d* in the range of 0.32–0.75). For the *overall* dimension: Both Interactive Help and Video Help were significantly higher than all other conditions except Intelligent Agent Help and Text Help, p <.05, *d* in the range of 0.26–0.58. No Help was lower than all other conditions, p <.05, *d* in the range of 0.38–0.58. See Figure 15.

RQ6: Does time spent on help facilities vary?

The Interactive Help, Intelligent Agent Help, and Video Help lead to longest time spent on help. In the All Help condition, participants spent the most time in the Interactive Help, and next longest in Video Help. In the All Help condition, participants are less likely to activate any help facility on load.

A one-way ANOVA found a significant effect of help facility condition on time spent on help at the p<.05 level [F(5,1640) = 36.85, p < 0.001, η_p^2 = 0.10]. Post-hoc testing using Tukey HSD found that participants in the Interactive Help (M=241, SD=313), Intelligent Agent Help (M=246, SD=382), and Video Help (M=238, SD=331), conditions spend more time on help than participants in Text Help (M=117, SD=198), and All Help (M=158, SD=202), p<.05, *d* in the range of 0.29–0.42.

A one-way within subjects ANOVA was conducted to compare time spent across different help facilities in the All Help condition. There was a significant difference in time spent, Wilk's λ = 0.929, F (3,275) = 7.04, p < .001, η_p^2 = 0.07. Post-hoc testing using a Bonferroni correction found that participants in the All Help condition spent significantly longer in the Interactive Help (M=63, SD=135) than in the Text Help (M=24, SD=81, *d*=0.35) and the Intelligent Agent Help (M=24, SD=86, *d*=0.34), p<.001. Time spent in Video Help was M=47, SD=144.

A one-way ANOVA found a significant effect of help facility condition on likelihood of startup help activation at the p<.05 level [F(5,1640) = 282.64, p < 0.001, η_p^2 = 0.46]. Post-hoc testing using Tukey HSD found that participants in the All Help condition

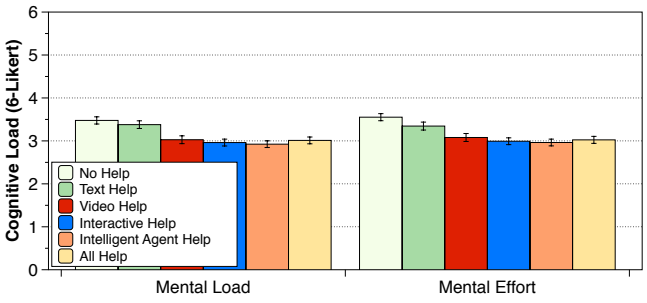

Figure 14: Means of 6-point Likert ratings for cognitive load (+/- SEM).

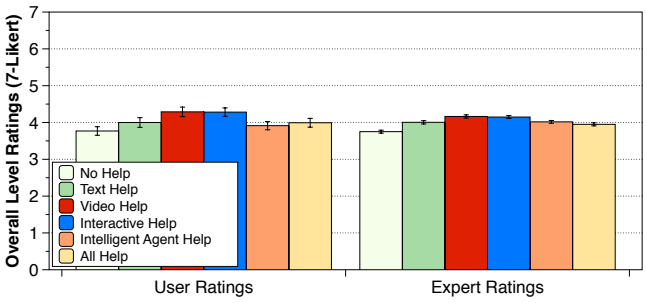

Figure 15: Mean 7-point Likert *overall* ratings for game levels (+/- SEM).

(M=66%) are significantly less likely to activate a help facility on startup than any other help facility condition, p<.05, *d* in the range of 0.20–0.58.

9 DISCUSSION

9.1 Game Making Help Facilities Are Crucial

The results show that Interactive Help promoted time spent, total editor activity, controls learnability, and learning motivation. Video Help promoted time spent, and learning motivation. These results highlight the important role that help facilities had in promoting motivated behavior, controls learnability, and learning motivation in *GameWorld*.

For cognitive load, No Help had the highest load with Text Help the second highest. All other conditions had lower cognitive load. These results show that help facilities can reduce the cognitive load for users, and that help facilities (e.g., Text Help) can have differential impacts.

Finally, results show that help facilities improve the quality of produced game levels. Both Interactive Help and Video Help led to the highest quality game levels, both self and expert rated. On the other hand, No Help led to the lowest quality. This demonstrates that help facilities improve the objective quality of produced artifacts in *GameWorld*.

9.2 Not All Help Facilities Are Made Equal

Results show that No Help is detrimental to most outcomes. Having some help facility was better than having no help facility. However, there was significant variance between help facilities. Text Help only marginally improved outcomes compared to No Help. Similarly, All Help and Intelligent Agent Help saw only marginal improvements compared to No Help, with the exception of cognitive load (on which both All Help and Intelligent Agent Help scored low). On the other hand, the results show that Interactive Help and Video Help led to significant improvements over No Help with medium–small effect sizes [21].

The results show that participants in the All Help condition are less likely to activate a help facility on startup. This potentially indicates too much choice, or a choice that was simply not meaningful [27, 28, 67, 104]. On the other hand, the two effective help facilities, Interactive Help and Video Help, are linear. These two help facilities are also the ones that participants in the All Help condition spent the most time with, suggesting that users preferred to spending time in these help facilities over Text Help and Intelligent Agent Help.

9.3 Why These Findings Occurred

Both Interactive Help and Video Help outperformed other conditions. Both Interactive Help and Video Help are able to moderate cognitive load in comparison to No Help and Text Help, through allowing users to follow guided step-by-step instructions. A reduction in cognitive load often results in better performance [89], which in this context translated to time spent, editor actions, controls learnability, learning motivation, and game level quality. The additional benefits of Interactive Help above and beyond other conditions in this study could be a result of performing actions immediately as they are being described during instruction. For example, decades of studies have shown the effectiveness of learning techniques that involve the act of doing while learning, including hands-on learning [98], active learning [110], and situated learning [70]. In a meta-analysis of 255 studies, active learning—which promotes directly interacting with learning material [13]—was shown to reduce failure rates in courses by 11% and increase student performance on course assessments by 0.47 standard deviations [110]. Therefore, interactive help facilities have a strong theoretical and empirical basis for their effectiveness. More work, however, is needed to understand why Intelligent Agent Help was less helpful than Interactive Help. It is possible that the number of dialog choices contained in the Intelligent Agent Help was overwhelming for users [27, 28, 67, 104]. More research is needed to understand how to best optimize different help facilities.

9.4 Recommendations for Game Making Software

Interactive Help and Video Help Improved Outcomes. Our results show that Interactive Help and Video Help lead to improved outcomes. However, in our review of game-making software, we found that while 89.4% had text documentation, only 52.9% had videos and 20.0% interactive tutorials. This indicates a potential missed opportunity for game-making software to better introduce systems to users[2].

Any Help Is Better Than No Help. Participants in No Help performed badly on all outcomes (time spent, controls learnability, learning motivation, cognitive load, total editor activity, and game

[2]One aspect not analyzed in this study is cost/ease of development, which may be a reason for Text Help's ubiquity.

level quality). Having some help facility was always better than having no help facility at all. This indicates that game-making software should always incorporate some form of help, even if simply basic documentation.

Be Wary of Giving Users Choice of Help. Results show that participants in the All Help condition, in which participants were able to choose which help facility to use, led to worse outcomes than Interactive Help or Video Help alone. Participants in All Help were less likely to activate help on startup than any other condition, and spent less time on help compared to Interactive Help, Video Help, and Intelligent Agent Help. This indicates that initially prompting the user with one good help facility will be more effective.

10 LIMITATIONS

Amazon Mechanical Turk (AMT) has been shown to be a reliable platform for experiments (e.g., [15, 76]). AMT workers also tend to represent a more diverse sample than the U.S. population [15, 19, 91]. However, future experiments restricted to experienced game developers could give more insight into help facilities' effects on experts. Despite that our AMT sample consists mainly of novices, these are likely the users who need the most scaffolding, and hence are an appropriate population to study in the context of help.

Longitudinal studies are needed. Although in the short-term, our results show that Interactive Help and Video Help are highly beneficial, the other help facilities could become more effective over time. For example, long-time users may find Text Help useful for looking up reference information. Longitudinal studies may determine, for example, that certain types of help are more appropriate for different levels of experience.

We took care to design help facilities in consultation with highly experienced game developers. Moreover, we ensured that game developers perceived the help facilities to be of high/similar quality, and that the help facilities were implemented similarly to other game-making software. Nevertheless, these help facilities could be constructed differently. For example, the intelligent agent could have been constructed to be similar to the user. A significant literature has shown that intelligent agents that are more similar to their users (termed similarity-attraction [16, 49]) along the dimensions of age, gender, race, clothing, etc. promote learning [7, 9, 11, 39, 51, 69, 96, 105]. Indeed, there are any number of changes to these help facilities that we can imagine. Nonetheless, there is value in contrasting the baseline developer-driven and developer-validated implementations here.

Finally, there are other forms of help that we are interested in. For example, providing template projects can be useful both as a starting point and for dissecting/understanding pre-built games. Additionally, we are interested in augmenting *GameWorld* with additional game genres (e.g., platformer, action RPG, etc.), and capabilities.

11 CONCLUSION

Game-making is increasingly pervasive, with an ever-larger number of game engines and software. With today's game-making software, it is increasingly becoming possible for novices and experts alike to create games. Nonetheless, game-making software is often complex. Help facilities can therefore play an important role in scaffolding knowledge in game-making. Results show that Interactive Help was the most promising help facility, leading to a greater positive impact on time spent, controls learnability, learning motivation, total editor activity, and game level quality. Video Help is a close second across these same measures. These results are directly relevant to designers, researchers, and developers, as they reveal how to best support novice game-making through help facilities. Future research in this domain can help cultivate the next generation of game-makers in an age where play and making are, more than ever, both ubiquitous and intertwined.

REFERENCES

[1] By Yasemin Allsop. 2016. A reflective study into children's cognition when making computer games. *British Journal of Educational Technology* (2016).

[2] Scott Ambler. 2012. Best Practices for Agile/Lean Documentation. *Agile Modeling* (2012).

[3] Erik Andersen, Eleanor O'Rourke, Yun-En Liu, Rich Snider, Jeff Lowdermilk, David Truong, Seth Cooper, and Zoran Popovic. 2012. The impact of tutorials on games of varying complexity. In *CHI*. https://doi.org/10.1145/2207676.2207687

[4] Keith Anderson, Elisabeth André, T. Baur, Sara Bernardini, M. Chollet, E. Chryssafidou, I. Damian, C. Ennis, A. Egges, P. Gebhard, H. Jones, M. Ochs, C. Pelachaud, Kaśka Porayska-Pomsta, P. Rizzo, and Nicolas Sabouret. 2013. The TARDIS framework: Intelligent virtual agents for social coaching in job interviews. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8253 LNCS (2013), 476–491. https://doi.org/10.1007/978-3-319-03161-3_35

[5] Rozalynd P. Anderson and Steven P. Wilson. 2009. Quantifying the effectiveness of interactive tutorials in medical library instruction. *Medical Reference Services Quarterly* 28, 1 (2009), 10–21. https://doi.org/10.1080/02763860802615815

[6] G. Antoniol, G. Canfora, A. De Lucia, and E. Merlo. 2003. Recovering code to documentation links in OO systems. (2003), 136–144. https://doi.org/10.1109/wcre.1999.806954

[7] Ivon Arroyo, Beverly Park Woolf, James M. Royer, and Minghui Tai. 2009. Affective gendered learning companions. In *Frontiers in Artificial Intelligence and Applications*, Vol. 200. 41–48. https://doi.org/10.3233/978-1-60750-028-5-41

[8] Batu Aytemiz, Isaac Karth, Jesse Harder, Adam M Smith, and Jim Whitehead. 2018. Talin : A Framework for Dynamic Tutorials Based on the Skill Atoms Theory. Aiide (2018), 138–144.

[9] Jeremy N. Bailenson, Jim Blascovich, and Rosanna E. Guadagno. 2008. Self-representations in immersive virtual environments. *Journal of Applied Social Psychology* 38, 11 (2008), 2673–2690.

[10] Amy Baylor. 1999. Intelligent agents as cognitive tools for education. *Educational technology* 39, 2 (1999), 36–40.

[11] Al Baylor and Yanghee Kim. 2004. Pedagogical agent design: The impact of agent realism, gender, ethnicity, and instructional role. *Intelligent Tutoring Systems* 1997 (2004), 592–603. https://doi.org/10.1007/978-3-540-30139-4_56

[12] Kelly Bergstrom, Jennifer Jenson, Emily Flynn-Jones, and Cristyne Hebert. 2018. Videogame Walkthroughs in Educational Settings: Challenges, Successes, and Suggestions for Future Use. *Proceedings of the 51st Hawaii International Conference on System Sciences* 9 (2018), 1875–1884. https://doi.org/10.24251/hicss.2018.237

[13] Charles C Bonwell and James A Eison. 1991. *Creating Excitement in the Classroom. WHAT IS ACTIVE LEARNING AND WHY IS IT IMPORTANT?* Technical Report.

[14] H David Brecht and Suzanne M Ogilby. 2008. Enabling a Comprehensive Teaching Strategy: Video Lectures. *Journal of Information Technology Education* 7 (2008), 71–86. http://go.galegroup.com/ps/i.do?action=interpret{&}id=GALE{%}7CA199685531{&}v=2.1{&}u=ggcl{&}it=r{&}p=AONE{&}sw=w{&}authCount=1

[15] Michael Buhrmester, Tracy Kwang, and Samuel D Gosling. 2011. Amazon's Mechanical Turk: A new source of inexpensive, yet high-quality, data? *Perspectives on psychological science* 6, 1 (2011), 3–5.

[16] Donn Byrne and Don Nelson. 1965. Attraction as a linear function of proportion of positive reinforcements. *Journal of Personality and Social Psychology* 1, 6 (1965), 659–663. https://doi.org/10.1037/h0022073

[17] Diane Carr. 2005. Contexts, gaming pleasures, and gendered preferences. *Simulation and Gaming* 36, 4 (2005), 464–482. https://doi.org/10.1177/1046878105282160

[18] John M Carroll, Penny L Smith-Kerker, James R Ford, and Sandra A Mazur-Rimetz. 1987. The Minimal Manual. *Human-Computer Interaction* 3, 2 (1987), 123–153. https://doi.org/10.1207/s15327051hci0302_2

[19] Jesse Chandler and Danielle Shapiro. 2016. Conducting clinical research using crowdsourced convenience samples. *Annual Review of Clinical Psychology* 12 (2016).

[20] Davida H. Charney and Lynne M. Reder. 1986. Designing Interactive Tutorials for Computer Users. *Human-Computer Interaction* 2, 4 (1986), 297–317. https://doi.org/10.1207/s15327051hci0204_2

[21] Jacob Cohen. 1992. A power primer. *Psychological bulletin* 112, 1 (1992), 155.

[22] Cristina Conati. 2002. Probabilistic assessment of user's emotions in educational games. *Applied Artificial Intelligence* 16, 7-8 (2002), 555–575. https://doi.org/10.1080/08839510290030390

[23] Cristina Conati and Xiaohong Zhao. 2004. Building and evaluating an intelligent pedagogical agent to improve the effectiveness of an educational game. (2004), 6. https://doi.org/10.1145/964442.964446

[24] Klaas Andries de Graaf. 2011. Annotating software documentation in semantic wikis. (2011), 5. https://doi.org/10.1145/2064713.2064718

[25] Sergio Cozzetti B. de Souza, Nicolas Anquetil, and Káthia M. de Oliveira. 2005. A study of the documentation essential to software maintenance. (2005), 68. https://doi.org/10.1145/1085313.1085331

[26] Jill Denner, Linda Werner, and Eloy Ortiz. 2012. Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers and Education* 58, 1 (2012), 240–249. https://doi.org/10.1016/j.compedu.2011.08.006

[27] Miriam Evans and Alyssa R. Boucher. 2015. Optimizing the power of choice: Supporting student autonomy to foster motivation and engagement in learning. *Mind, Brain, and Education* 9, 2 (2015), 87–91. https://doi.org/10.1111/mbe.12073

[28] Terri Flowerday and Gregory Schraw. 2000. Teacher beliefs about instructional choice: A phenomenological study. *Journal of Educational Psychology* 92, 4 (2000), 634–645. https://doi.org/10.1037/0022-0663.92.4.634

[29] Vera Frith, Jacob Jaftha, and Robert Prince. 2004. Evaluating the effectiveness of interactive computer tutorials for an undergraduate mathematical literacy course. *British Journal of Educational Technology* 35, 2 (2004), 159–171.

[30] Julian Frommel, Kim Fahlbusch, Julia Brich, and Michael Weber. 2017. The Effects of Context-Sensitive Tutorials in Virtual Reality Games. (2017), 367–375. https://doi.org/10.1145/3116595.3116610

[31] Vihanga Gamage and Cathy Ennis. 2018. Examining the effects of a virtual character on learning and engagement in serious games. (2018), 1–9. https://doi.org/10.1145/3274247.3274499

[32] Jacqueline Gaston and Seth Cooper. 2017. To Three or not to Three: Improving Human Computation Game Onboarding with a Three-Star System. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17* (2017), 5034–5039. https://doi.org/10.1145/3025453.3025997

[33] Elisabeth R. Gee and Kelly M. Tran. [n.d.]. *Video Game Making and Modding*. 238–267 pages. https://doi.org/10.4018/978-1-4666-8310-5.ch010

[34] James Paul Gee. 2005. Learning by Design: Good Video Games as Learning Machines. *E-Learning and Digital Media* 2, 1 (2005), 5–16. https://doi.org/10.2304/elea.2005.2.1.5 arXiv:arXiv:1011.1669v3

[35] Gómez-Martín, Marco A., Pedro P. Gómez-Martín and Pedro A. González-Calero. 2004. Game-driven intelligent tutoring systems. In *International Conference on Entertainment Computing*. Springer, 108–113.

[36] Michael Cerny Green, Ahmed Khalifa, Gabriella A. B. Barros, Tiago Machado, Andy Nealen, and Julian Togelius. 2018. AtDelfi: Automatically Designing Legible, Full Instructions For Games. In *FDG*. arXiv:1807.04375 http://arxiv.org/abs/1807.04375

[37] Michael Cerny Green, Ahmed Khalifa, Gabriella A. B. Barros, and Julian Togelius. 2018. "Press Space to Fire": Automatic Video Game Tutorial Generation. (2018), 75–80. arXiv:1805.11768 http://arxiv.org/abs/1805.11768

[38] Jeffrey Greene and Laura Palmer. 2011. It's all in the game: Technical communication's role in game documentation. *Intercom* 3, 63 (2011), 6–9.

[39] Rosanna E Guadagno, Jim Blascovich, Jeremy N Bailenson, and Cade McCall. 2007. Virtual humans and persuasion: The effects of agency and behavioral realism. *Media Psychology* 10, 1 (2007), 1–22. https://doi.org/10.1080/15213260701300865

[40] Carl Gutwin, Rodrigo Vicencio-Moreira, and Regan L Mandryk. 2016. Does Helping Hurt?: Aiming Assistance and Skill Development in a First-Person Shooter Game. *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play* (2016), 338–349. https://doi.org/10.1145/2967934.2968101

[41] Erik Harpstead, Brad A. Myers, and Vincent Aleven. 2013. In search of learning: Facilitating data analysis in educational games. In *Conference on Human Factors in Computing Systems - Proceedings*. https://doi.org/10.1145/2470654.2470667

[42] Casper Harteveld and Steven Sutherland. 2015. The Goal of Scoring: Exploring the Role of Game Performance in Educational Games. *Proceedings of the 33rd annual ACM conference on Human factors in computing systems (CHI 2015)* (2015).

[43] Elisabeth R. Hayes and Ivan Alex Games. 2008. Making Computer Games and Design Thinking. *Games and Culture* 3, 3-4 (2008), 309–332. https://doi.org/10.1177/1555412008317312

[44] Elisabeth R. Hayes and Ivan Alex Games. 2008. Making computer games and design thinking: A review of current software and strategies. *Games and Culture* 3, 3-4 (2008), 309–332. https://doi.org/10.1177/1555412008317312

[45] Kate Howland and Judith Good. 2015. Learning to communicate computationally with Flip: A bi-modal programming language for game creation. *Computers and Education* 80 (2015), 224–240. https://doi.org/10.1016/j.compedu.2014.08.014

[46] Johan Huizinga. 2014. *Homo Ludens Ils 86*. Routledge.

[47] Gwo-Jen Hwang and Hsun-Fang Chang. 2011. A formative assessment-based mobile learning approach to improving the learning attitudes and achievements of students. *Computers & Education* 56, 4 (2011), 1023–1031.

[48] Ioanna Iacovides, Anna L. Cox, and Thomas Knoll. 2014. Learning the Game: Breakdowns, Breakthroughs and Player Strategies. *Proceedings of the extended abstracts of the 32nd annual ACM conference on Human factors in computing systems - CHI EA '14* (2014), 2215–2220. https://doi.org/10.1145/2559206.2581304

[49] Katherine Isbister and Clifford Nass. 2000. Consistency of personality in interactive characters: verbal cues, non-verbal cues, and user characteristics. *International Journal of Human-Computer Studies* 53 (2000), 251–267. https://doi.org/10.1006/ijhc.2000.0368

[50] Colby Johanson and Regan L. Mandryk. 2016. Scaffolding Player Location Awareness through Audio Cues in First-Person Shooters. (2016), 3450–3461. https://doi.org/10.1145/2858036.2858172

[51] Amy M. Johnson, Matthew D. Didonato, and Martin Reisslein. 2013. Animated agents in K-12 engineering outreach: Preferred agent characteristics across age levels. *Computers in Human Behavior* 29, 4 (2013), 1807–1815.

[52] Yasmin B Kafai. 2006. Playing and making games for learning: Instructionist and constructionist perspectives for game studies. *Games and Culture* 1, 1 (2006), 36–40. https://doi.org/10.1177/1555412005281767

[53] Yasmin B. Kafai and Quinn Burke. 2015. Constructionist Gaming: Understanding the Benefits of Making Games for Learning. *Educational Psychologist* 50, 4 (2015), 313–334. https://doi.org/10.1080/00461520.2015.1124022

[54] Verena Käfer, Daniel Kulesz, and Stefan Wagner. 2017. What Is the Best Way For Developers to Learn New Software Tools? *The Art, Science, and Engineering of Programming* 1, 2 (2017). https://doi.org/10.22152/programming-journal.org/2017/1/17

[55] Dominic Kao. 2019. Exploring the Effects of Growth Mindset Usernames in STEM Games. *American Education Research Association* (2019).

[56] Dominic Kao. 2019. JavaStrike: A Java Programming Engine Embedded in Virtual Worlds. In *Proceedings of The Fourteenth International Conference on the Foundations of Digital Games*.

[57] Dominic Kao. 2019. The Effects of Anthropomorphic Avatars vs. Non-Anthropomorphic Avatars in a Jumping Game. In *The Fourteenth International Conference on the Foundations of Digital Games*.

[58] Dominic Kao. 2020. The effects of juiciness in an action RPG. *Entertainment Computing* 34, November 2018 (2020), 100359. https://doi.org/10.1016/j.entcom.2020.100359

[59] Dominic Kao and D. Fox Harrell. 2015. Exploring the Impact of Role Model Avatars on Game Experience in Educational Games. *The ACM SIGCHI Annual Symposium on Computer-Human Interaction in Play (CHI PLAY)* (2015).

[60] Dominic Kao and D. Fox Harrell. 2015. Mazzy: A STEM Learning Game. *Foundations of Digital Games* (2015).

[61] Dominic Kao and D Fox Harrell. 2016. Exploring the Effects of Dynamic Avatars on Performance and Engagement in Educational Games. In *Games+Learning+Society (GLS 2016)*.

[62] Dominic Kao and D. Fox Harrell. 2016. Exploring the Effects of Encouragement in Educational Games. *Proceedings of the 34th Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI 2016)* (2016).

[63] Dominic Kao and D. Fox Harrell. 2016. Exploring the Impact of Avatar Color on Game Experience in Educational Games. *Proceedings of the 34th Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI 2016)* (2016).

[64] Dominic Kao and D. Fox Harrell. 2017. MazeStar: A Platform for Studying Virtual Identity and Computer Science Education. In *Foundations of Digital Games*.

[65] Dominic Kao and D. Fox Harrell. 2017. Toward Understanding the Impact of Visual Themes and Embellishment on Performance, Engagement, and Self-Efficacy in Educational Games. *The annual meeting of the American Educational Research Association (AERA)* (2017).

[66] Dominic Kao and D. Fox Harrell. 2018. The Effects of Badges and Avatar Identification on Play and Making in Educational Games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI'18*.

[67] Idit Katz and Avi Assor. 2007. When choice motivates and when it does not. *Educational Psychology Review* 19, 4 (2007), 429–442. https://doi.org/10.1007/s10648-006-9027-y

[68] Caitlin Kelleher and Randy Pausch. 2005. Stencils-Based Tutorials: Design and Evaluation. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '05* (2005), 541. https://doi.org/10.1145/1054972.1055047

[69] Yanghee Kim and Amy L. Baylor. 2006. Pedagogical agents as learning companions: The role of agent competency and type of interaction. *Educational Technology Research and Development* 54, 3 (2006), 223–243.

[70] Jean Lave and Etienne Wenger. 1991. Situated learning: Legitimate peripheral participation. *Learning in doing* 95 (1991), 138. https://doi.org/10.2307/2804509 arXiv:arXiv:1011.1669v3

[71] Jonathan Lazar, Adam Jones, and Ben Shneiderman. 2006. Workplace user frustration with computers: An exploratory investigation of the causes and

severity. *Behaviour & Information Technology* 25, 03 (2006), 239–251.
[72] James C. Lester, Sharolyn a. Converse, Susan E. Kahler, S. Todd Barlow, Brian a. Stone, and Ravinder S. Bhogal. 1997. The Persona Effect: Affective Impact of Animated Pedagogical Agents. *Proceedings of the SIGCHI conference on Human Factors in Computing Systems - CHI '97* (1997), 359–366. https://doi.org/10.1145/258549.258797
[73] Gillian Lieberman, Richard Abramson, Kevin Volkan, and Patricia J. McArdle. 2002. Tutor versus computer: A prospective comparison of interactive tutorial and computer-assisted instruction in radiology education. *Academic Radiology* 9, 1 (2002), 40–49. https://doi.org/10.1016/S1076-6332(03)80295-7
[74] Dennis K. Lieu. 1999. Using interactive multimedia computer tutorials for engineering Graphics education. *Journal for Geometry and Graphics* 3, 1 (1999), 85–91.
[75] Eric Malbos, Ronald M. Rapee, and Manolya Kavakli. 2013. Creation of interactive virtual environments for exposure therapy through game-level editors: Comparison and tests on presence and anxiety. *International Journal of Human-Computer Interaction* (2013). https://doi.org/10.1080/10447318.2013.796438
[76] Winter Mason and Siddharth Suri. 2012. Conducting behavioral research on Amazon's Mechanical Turk. *Behavior Research Methods* 44, 1 (2012), 1–23. https://doi.org/10.3758/s13428-011-0124-6 arXiv:/ssrn.com/abstract=1691163 [http:]
[77] Richard E Mayer. 2002. Multimedia learning. In *Psychology of learning and motivation*. Vol. 41. Elsevier, 85–139.
[78] Richard E Mayer. 2006. Ten research-based principles of multimedia learning. *Web-based learning: Theory, research, and practice* (2006), 371–390.
[79] Richard E. Mayer and Richard B. Anderson. 1991. Animations Need Narrations: An Experimental Test of a Dual-Coding Hypothesis. *Journal of Educational Psychology* 83, 4 (1991), 484–490. https://doi.org/10.1037/0022-0663.83.4.484
[80] Barbara Mirel. 1998. "Applied Constructivism" for User Documentation: Alternatives to Conventional Task Orientation. *Journal of Business and Technical Communication* 12, 1 (1998), 7–49. https://doi.org/10.1177/1050651998012001002
[81] Ryan M Moeller and Others. 2016. *Computer games and technical communication: Critical methods and applications at the intersection*. Routledge.
[82] Raphaël Moirn, P.-M. Léger, Sylvain Senecal, M.-C.B. Roberge, Mario Lefebvre, and Marc Fredette. 2016. The effect of game tutorial: A comparison between casual and hardcore gamers. *CHI PLAY 2016 - Proceedings of the Annual Symposium on Computer-Human Interaction in Play Companion* (2016), 229–237. https://doi.org/10.1145/2968120.2987730
[83] Andreea Molnar and Patty Kostkova. 2013. If you build it would they play ? Challenges and Solutions in Adopting Health Games for Children. *CHI 2003 Workshop: Let's talk about Failures: Why was the Game for Children not a Success?* June (2013), 9–12.
[84] Andreea Molnar and Patty Kostkova. 2014. Gaming to master the game: Game usability and game mechanics. *SeGAH 2014 - IEEE 3rd International Conference on Serious Games and Applications for Health, Books of Proceedings* May (2014). https://doi.org/10.1109/SeGAH.2014.7067091
[85] Niklas Nylund. 2015. Walkthrough and let's play: evaluating preservation methods for digital games. In *Proceedings of the 19th International Academic Mindtrek Conference*. ACM, 55–62.
[86] E O'Rourke, Kyla Haimovitz, Christy Ballweber, Carol S. Dweck, and Zoran Popović. 2014. Brain points: a growth mindset incentive structure boosts persistence in an educational game. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14* (2014), 3339–3348. http://dl.acm.org/citation.cfm?id=2557157
[87] Eleanor O'Rourke, Erin Peach, Carol S Dweck, and Zoran Popovi. 2016. Brain Points: A Deeper Look at a Growth Mindset Incentive Structure for an Educational Game. *Proceedings of the Third ACM Conference on Learning@Scale* (2016), 41–50. https://doi.org/10.1145/2876034.2876040 arXiv:arXiv:1011.1669v3
[88] Fred G Paas. 1992. Training strategies for attaining transfer of problem-solving skill in statistics: A cognitive-load approach. *Journal of educational psychology* 84, 4 (1992), 429.
[89] Fred G.W.C. Paas. 1992. Training Strategies for Attaining Transfer of Problem-Solving Skill in Statistics: A Cognitive-Load Approach. *Journal of Educational Psychology* (1992). https://doi.org/10.1037/0022-0663.84.4.429
[90] Claus Pahl. 2002. An Evaluation of Scaffolding for Virtual Interactive Tutorials. *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education* (2002), 740–746. http://www.editlib.org/p/15295
[91] Gabriele Paolacci, Jesse Chandler, and Panagiotis G Ipeirotis. 2010. Running experiments on amazon mechanical turk. (2010).
[92] S Papert and Idit Harel. 1991. Situating Constructionism. *Constructionism* (1991).
[93] Cécile Paris and Keith Vander Linden. 2007. Building knowledge bases for the generation of software documentation. (2007), 734. https://doi.org/10.3115/993268.993296 arXiv:9607026 [cmp-lg]
[94] Clara-inés Pena, J L Marzo, and Josep-lluis De Rosa. 2002. Intelligent Agents in a Teaching and Learning Environment on the Web. *Proceedings of the International Conference on Advanced Learning Technologies* (2002), 21–27.
[95] Jean Piaget. 2013. *Play, dreams and imitation in childhood*. https://doi.org/10.4324/9781315009698
[96] Jean A. Pratt, Karina Hauser, Zsolt Ugray, and Olga Patterson. 2007. Looking at human-computer interface design: Effects of ethnicity in computer agents. *Interacting with Computers* 19, 4 (2007), 512–523.
[97] Sheri Graner Ray. 2010. Tutorials: Learning to play. *Gamasutra, http://www.gamasutra. com/view/feature/134531/tutorials_learning_to_play. php* (2010).
[98] Melissa Regan and Sheri Sheppard. 1996. Interactive multimedia courseware and the hands-on learning experience: an assessment study. *Journal of engineering education* 85, 2 (1996), 123–132.
[99] Lloyd P Rieber. 1994. *Computers graphics and learning*. Brown & Benchmark Pub.
[100] Judy Robertson. 2012. Making games in the classroom: Benefits and gender concerns. *Computers and Education* 59, 2 (2012), 385–398. https://doi.org/10.1016/j.compedu.2011.12.020
[101] Judy Robertson. 2013. The influence of a game-making project on male and female learners' attitudes to computing. *Computer Science Education* 23, 1 (2013), 58–83. http://10.0.4.56/08993408.2013.774155{%}5Cnhttp://search.ebscohost.com/login.aspx?direct=true{&}db=eue{&}AN=87373893{&}site=ehost-live
[102] Judy Robertson and Cathrin Howells. 2008. Computer game design: Opportunities for successful learning. *Computers and Education* 50, 2 (2008), 559–578. https://doi.org/10.1016/j.compedu.2007.09.020
[103] David Rojas, Rina R. Wehbe, Lennart E. Nacke, Matthias Klauser, Bill Kapralos, and Dennis L. Kappen. 2013. EEG-based assessment of video and in-game learning. (2013), 667. https://doi.org/10.1145/2468356.2468474
[104] David H Rose and Anne Meyer. 2002. Teaching Every Student in the Digital Age: Universal Design for Learning. *Ericedgov* (2002), 216. https://doi.org/10.1007/s11423-007-9056-3
[105] Rinat B. Rosenberg-Kima, E. Ashby Plant, Celestee E. Doerr, and Amy Baylor. 2010. The influence of computer-based model's race and gender on female students' attitudes and beliefs towards engineering. *Journal of Engineering Education* (2010), 35–44. https://doi.org/10.1002/j.2168-9830.2010.tb01040.x
[106] Richard M. Ryan, C. Scott Rigby, and Andrew Przybylski. 2006. The Motivational Pull of Video Games: A Self-Determination Theory Approach. *Motivation and Emotion* 30, 4 (2006), 344–360. https://doi.org/10.1007/s11031-006-9051-8
[107] Katie Salen and Eric Zimmerman. 2004. *Rules of Play: Game Design Fundamentals*. 670 pages. https://doi.org/10.1093/intimm/dxs150 arXiv:arXiv:1011.1669v3
[108] Ted Selker. 2002. COACH: a teaching agent that learns. *Commun. ACM* (2002). https://doi.org/10.1145/176789.176799
[109] Aviv Shachak, Rustam Dow, Jan Barnsley, Karen Tu, Sharon Domb, Alejandro R Jadad, and Louise Lemieux-Charles. 2013. User Manuals for a Primary Care Electronic Medical Record System: A Mixed-Methods Study of User- and Vendor-Generated Documents. *IEEE Transactions on Professional Communication* 56, 3 (2013), 194–209. https://doi.org/10.1109/tpc.2013.2263649
[110] Mel Silberman. 1996. *Active Learning: 101 Strategies To Teach Any Subject*. ERIC.
[111] Katherine Stiwinter. 2013. Using an interactive online tutorial to expand library instruction. *Internet Reference Services Quarterly* 18, 1 (2013), 15–41.
[112] John Sweller. 1988. Cognitive load during problem solving: Effects on learning. *Cognitive Science* (1988). https://doi.org/10.1016/0364-0213(88)90023-7
[113] John Sweller, Jeroen J G Van Merrienboer, and Fred G W C Paas. 1998. Cognitive architecture and instructional design. *Educational psychology review* 10, 3 (1998), 251–296.
[114] Philip W Tiemann and Susan M Markle. 1990. Effects of varying interactive strategies provided by computer-based tutorials for a software application program. *Performance Improvement Quarterly* 3, 2 (1990), 48–64.
[115] Barbara Tversky, Julie Bauer Morrison, and Mireille Betrancourt. 2002. Animation: can it facilitate? *Int. J. Human-Computer Studies Schnotz & Kulhavy* 57 (2002), 247–262. https://doi.org/10.1006/ijhc.1017 arXiv:arXiv:1011.1669v3
[116] Unity. 2019. Unity Public Relations. https://unity3d.com/public-relations
[117] Hans van der Meij and Jan Van Der Meij. 2014. A comparison of paper-based and video tutorials for software learning. *Computers & education* 78 (2014), 150–159.
[118] Monica Visani Scozzi, Ioanna Iacovides, and Conor Linehan. 2017. A Mixed Method Approach for Evaluating and Improving the Design of Learning in Puzzle Games. (2017), 217–228. https://doi.org/10.1145/3116595.3116628
[119] Xiaobo Wang, Guanhui Lai, and Chao Liu. 2009. Recovering Relationships between Documentation and Source Code based on the Characteristics of Software Engineering. *Electronic Notes in Theoretical Computer Science* 243 (2009), 121–137. https://doi.org/10.1016/j.entcs.2009.07.009
[120] Helen Wauck and Wai-Tat Fu. 2017. A Data-Driven, Multidimensional Approach to Hint Design in Video Games. (2017), 137–147. https://doi.org/10.1145/3025171.3025224
[121] Megan A. Winget and Wiliam Walker Sampson. 2011. Game development documentation and institutional collection development policy. (2011), 29. https://doi.org/10.1145/1998076.1998083
[122] Alyssa Friend Wise and Kevin O'Neill. 2009. Beyond more versus less: A reframing of the debate on instructional guidance. In *Constructivist Instruction: Success or Failure?* https://doi.org/10.4324/9780203878842
[123] Michael Wooldridge and Nicholas R Jennings. 1995. Intelligent agents: Theory and practice. *The knowledge engineering review* (1995). https://doi.org/10.1017/

S0269888900008122

[124] George J Xeroulis, Jason Park, Carol-Anne Moulton, Richard K Reznick, Vicki LeBlanc, and Adam Dubrowski. 2007. Teaching suturing and knot-tying skills to medical students: a randomized controlled study comparing computer-based video instruction and (concurrent and summary) expert feedback. *Surgery* 141, 4 (2007), 442–449.

[125] Mark Zachry. 2001. Constructing usable documentation: A study of communicative practices and the early uses of mainframe computing in industry. *Journal of technical writing and communication* 31, 1 (2001), 61–76.

[126] Junji Zhi, Vahid Garousi-Yusifoğlu, Bo Sun, Golara Garousi, Shawn Shahnewaz, and Guenther Ruhe. 2015. Cost, benefits and quality of software development documentation: A systematic mapping. *Journal of Systems and Software* 99, January (2015), 175–198. https://doi.org/10.1016/j.jss.2014.09.042

[127] Martina Ziefle. 1998. Effects of Display Resolution on Visual Performance. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 40, 4 (1998), 554–568. https://doi.org/10.1518/001872098779649355