2020

The Gamification of Code: Programming Through Play in Blended Classrooms

Kristin Carlson
Illinois State University

Rick Valentin
Illinois State University

Chapter 5
The Gamification of Code: Programming Through Play in Blended Classrooms

Kristin Carlson
Illinois State University, USA

Rick Valentin
Illinois State University, USA

ABSTRACT

Teaching an introductory web design course is already a blended environment. Students meet face-to-face, yet have access to a myriad of online resources, YouTube videos, blogs, and forums to support their learning. However, the challenges of learning to understand code can inhibit students and diminish their motivation to look for resources. The authors have attempted to address this issue by focusing on the use and design of games for learning to code, as well as providing video lecture material in combination with the traditional face-to-face learning environment. By using games and gamification in the course design, the authors have found that students are able not only to bridge their knowledge between modalities more smoothly, but that they understand that there are multiple ways to solve a problem and feel empowered to search for solutions in innovative ways.

INTRODUCTION

In some senses, computer programming itself is one of the best computer games of all. In the "computer programming game," there are obvious goals and it is easy to generate more" (Malone, 1980, pg.7).

Learning to code can be challenging as it requires students learn new ways of thinking and processing information. Challenges can be particularly strong in scenarios where students are at many levels to begin with, and may have different learning preferences and paces. There are currently many resources available to help students learn to code—including books, blogs, online video tutorials, online code camps

DOI: 10.4018/978-1-7998-0242-6.ch005

and competitions, forums where they can ask questions, and repositories of existing code they can use to start with. However, there are not as many resources that use games and playful approaches to teach code.

In our University level Creative Technologies Program, the beginning web design class is the first coding class students take. Many of them are scared of coding because it sounds like math; using abstract concepts to learn "code"—a hidden, mystical set of concepts that are not easily grasped. However, once students start diving in, the common response we see is that they "get it" much more quickly than they expected, are proud of themselves, and are motivated to explore more deeply. Performance throughout the course is quite high and students become excited about the content, but individual students grasp the course concepts at different paces.

We decided to explore the application of games to the learning process for a few reasons: multisensory approaches in playful situations have been shown to support the comprehension of abstract concepts (Katai and Toth, 2010); the combination of multisensory and playful situations, with online reference material, could support the variety of students' learning paces; and, these playful situations encourage collaborative and competitive learning.

Our current students prefer to learn through video tutorials, or following along with an instructor as opposed to reading resources. Although the approach of follow-along programming can encourage active learning, the nature of code can still be difficult and relies heavily on concepts, memorization, and practice. We have found that the logic and procedures of coding are similar to those in gameplay and creative processes, and can be made transferable if the points are created, especially across mediums. We wondered how students would bridge the two forms if they could map the structure of code from one of play to one of learning. To leverage this as a blended learning classroom, we developed multiple games over a variety of platforms (traditional and online) to support a gameful way of teaching code, in addition to using existing games. Some games are played individually, some in small groups, and some in large groups to include a variety of approaches.

We used several games for teaching, including Simon Says with HTML, a CSS Scavenger Hunt, a computer pathfinding game, and Jeopardy. This chapter will highlight the games we designed for teaching specific concepts and the existing games we found that support relevant concepts. We will discuss our assessment of using the games across in-person and online platforms, in both individual and group situations, and how students felt these learning opportunities supported their experience.

BACKGROUND

Blended Learning as a Multisensory Process

With the rise of accessible technology and students coming in to the classroom as digital natives, blended learning has been emerging as a teaching strategy that supports both engagement and efficiency (Bonk and Graham, 2006). Some approaches focus on the effective use of blended learning for corporate job training situations (e.g., flight simulations) (Bonk and Graham, 2006), but term describes meaningfully constructed learning environments that leverage both face-to-face and technological options (Katai and Toth, 2010). Blended learning is noted as a way to combine the best of face-to-face and online learning (Bourne and Seaman, 2005), and also emphasizing that a thoughtful fusion is important to provide more engaging learning experiences (Garrison and Vaughan, 2008). In higher education, interaction and collaboration-based learning experiences are noted as supporting more complex knowledge acqui-

sition (Garrison and Vaughan, 2008). Traditional university learning situations may include strategies for adding online discussion to shift the focus and engagement of students (Han and Ellis, 2019), or the use of video content to support student's learning pace and ability to refer back to content (Bonk and Graham, 2006). Implementing a "flipped" classroom that puts active learning in class time and moving traditional instructional content online can provide a better perception of the learning environment (Baepler et al., 2014).

Blended learning provides an opportunity to apply a multisensory and experiential approach to assist learning. Gardner, a developmental psychologist, discussed nine intelligences (including Interpersonal, Bodily-kinesthetic, and Linguistic intelligences) and notes that students need to learn in a variety of ways (2000). Multisensory learning refers to how students use a variety of senses to support their engagement in learning, mixing visual, auditory, tactile, and kinesthetic approaches (Shams and Seitz, 2008). From a young age, children are learning through combined senses (for example, kinesthetic and sound) to develop their perception. Although most multisensory research has been done on specific stimuli in highly controlled setting, there is consensus that a focus on unisensory learning (sticking to a single sensory modality, such as listening to a lecture) does not support learning as well as multisensory methods (Shams and Seitz, 2008). Because teaching through multisensory environments mimics natural settings (Shams and Seitz, 2008) and uses a variety of modalities to support a variety of learners (Katai and Toth, 2010), blended learning using these modalities across media provides increased access and flexibility for learners (Bonk and Graham, 2006). Programming in particular is noted as being abstract, which contributes to student's perceptions that it is difficult to learn (Navrat, 1994). The concepts of loops and recursion, a method that will repeat an action numerous times with a minimal amount of code, are considered to be an especially high level of abstraction that can be better understood through playful multisensory approaches, as explored by Katai and Toth when they developed dance approaches to learning loop structures (2010).

Experiential learning refers to how a student can actively participate in their learning ("learning by doing") through direct context and application (Kolb and Fry, 1974). Kolb also notes that knowledge develops by "the combination of grasping and transforming experience" (1974 p. 41). Learning through play and experimentation with abstract concepts was discussed by Dewey in the early 20th century (Dewey, 1916; Farber, 2018). The learning of abstract concepts such as math, algorithms, or programming can be made more concrete through playful experiential and multisensory methods, because it reframes the information in ways that can be more tangible to the learner (Dewey, 1916; Farber, 2018; Katai and Toth, 2010).

The Role of Games in Learning

We discuss the role of games in education from a perspective of play, curiosity, and motivation, rather than from whether digital interactions support learning differently than in-person learning. Games have existed in learning for decades, as Dewey describes experiential learning as often playful and intrinsically motivating (1916) and Piaget discusses play "as the work of children" (Cohen, 2007). Sutton-Smith observes that babies don't play, they just explore and learn (1986). This suggests that children are not separating play from their learning. Gamification has long been a component of learning because making work "playful" has been noted as making learning more engaging (Kapp, 2012). For example, in music classes, a variety of music games are played as a way to teach children rhythmic concepts (Turner, 2004). Serious games are noted as specifically designed games to support the learning of particular content, but

gamification is noted as "a careful and considered application of game thinking to solving problems and encouraging learning using all the elements of games that are appropriate" (Kapp, 2012). Although the term "serious games" was not coined until 2002, the importance of play in learning has been apparent since Plato (Wilkinson, 2016) and is an integral facet of exploring the world.

Games are situations where players willingly learn, take on challenges, explore, experiment, and work collaboratively because they are motivated (Kapp, 2012). "Playing a game is the voluntary attempt to overcome unnecessary obstacles" (Suits, 2005, p. 55). Games are play, where motivated people use their curiosity to engage with new situations. According to Gee in his book *What Video Games Have to Teach Us About Learning and Literacy*, motivation is the most important factor in learning, and learning will stop when the motivation stops. Focusing on how to create and sustain motivation is the most important part of teaching, whether it is dependent on a game or not (Gee, 2003). The conflict happens between levels of ability: games often operate by challenging the top players while classrooms are trying to engage the students furthest behind (Gee, 2003). However, the combination of games and blended learning could make teaching more accessible and facilitate a variety of learning levels (Gee, 2007). Philpot, Hall, Hubing, & Flori (2005) noted that "special multimedia applications and computer-games can increase students' motivation to learn, and often lead to a better understanding of the studied topics." Technology is noted as useful tool in teaching by "allowing students to design, explore, experiment, and model complex and abstract phenomena" (American Council on Education in Katai and Toth, 2010, p. 245). The difference between 2D and 3D games has been explored in relation to this, to see whether there is a difference in student learning based on the resolution of the technology. It was found that 2D environments were more successful in retention because they were using a lower resolution of information, which required less cognitive load in the students than the high-resolution 3D environments (though the 3D environments found a higher level of student presence in the moment) (Schrader and Bastiaens, 2012). However, the content in real-time strategy games such as Age of Mythology and Age of Empires lends itself to provokinf curiosity in students to explore real-world content on their own, outside of the game, to teach connections between history and society (Gee, 2003). Balance in the games and in the teaching is important (Farber, 2018). "If a learning environment is well constructed, then the students will like the learning environment. The sub-factors of likeability can be specified as engagement, challenge, concentration, goal clarity, feedback, autonomy, preference and immersion" (Ak and Kutlu, 2017). Overall, games tend to be viewed as beneficial to learning when they are used thoughtfully and include instructional design in the process, not just throwing in a game that is unrelated to make a challenging topic more fun (Ak and Kutlu, 2017; Dicheva et al., 2015; Farber, 2018; Kapp, 2012; Tanenbaum, Gardner, & Cowling, 2017).

Malone defines "fun" in educational games in three categories; challenge, fantasy, and curiosity (1980). In challenge, game designers need to identify appropriate goals for the learners, then apply uncertain outcomes, hidden information, and elements of randomness to engage learners in a challenge that is unpredictable and sparks their curiosity. Fantasy is categorized as extrinsic and intrinsic fantasy: extrinsic fantasy is based on how the player's skills are correctly applied to win the game, but in intrinsic fantasy the player's skills are based on the fantasy, and the fantasy is also based on the skills. For example, extrinsic fantasy would race a car based on how quickly the player answers math questions correctly, whereas intrinsic fantasy would use physical actions of throwing darts to determine distances between objects (Malone, 1980). Curiosity creates the motivation to learn, based on what the player can do and comprehend, and is defined as sensory or cognitive. This paper gives a comprehensive approach to designing games for engagement, and could be compared to McCarthy McCarthy, Wright, Wallace, &

Dearden's work on designing for enchantment (2006). This work suggests guidelines such as designing for sensuousness; a sense of being in play; inclusion of paradox, openness, and ambiguity in the design; and the opportunity for transformational experiences. Concepts of paradox, openness, and ambiguity can relate to challenge in Malone's work, as designing for sensuousness and transformational characters of experience can relate to sensory and cognitive curiosity. These parallels suggest how important aspects of goals, unpredictability, and sensory engagement can be in aspects of play and engaged learning.

Bellotti, Kapralos, Lee, Moreno-Ger, & Berta (2013) state that "many educational games do not properly translate knowledge, facts, and lessons into the language of games. This results in games that are often neither engaging nor educational," which suggests that games are often applied to learning in haphazard ways to make it seem more engaging. In Kapp's book, *The Gamification of Learning and Instruction*, he provides a meta-analysis of six survey papers that evaluate the success of games in learning from 1992–2011. More than 50% of all studies showed benefits of games in learning; the way that games were implemented, how the studies were run, and what content was being taught all contributed to the effectiveness of the learning. Although student performance was noted as one method of evaluation, most of the studies focused on whether students met the learning outcomes. Some highlights include: learning was more successful when specific content was addressed; games provided better attitudes toward learning and increased student motivation; well-designed games did not require effort to learn, but could focus on the content; visual realism and entertainment values were not important to learning; and games supported "higher-order thinking such as planning and reasoning more than factual or verbal knowledge" (p. 102). A review paper by de Frietas states: "To the question: are games effective learning tools, the answer from the research is overwhelmingly positive" (2018, p. 80). Though to get deeper, de Frietas also notes that it takes work to properly align learning outcomes with game playability for solid learning, and that proper feedback systems and cross-disciplinary approaches in social environments is important to make the use of games in learning successful (2018).

Examples of Games to Teach and Support Programming

Although there is an emphasis in the serious games discipline on the use of games in education, there is little research specifically discussing the use of games to teaching programming. What is studied more is the use of games to teach abstract concepts, when applied to math, algorithms, and programming. Kapp notes:

Although research has shown that some games can provide effective learning for a variety of learners for several different tasks (e.g., math, attitudes, electronics, and economics), this does not tell us whether to use a game for our specific instructional task. We should not generalize from research on the effectiveness of one game in one learning area for one group of learners to all games in all learning areas for all learners (2012).

One concept noted specifically by Kapp in his 2012 meta-analysis of six survey papers on games in education is the principle of uncertainty, particularly when applied to games teaching math, where uncertain rewards in the game (not to be confused with unclear rules or content) supported better student engagement and attitude than certain rewards. Overall, games are used to teach programming, but the focus tends to be more on student motivation and engagement rather than on test scores and performance

overall. Games may not be more effective at teaching programming themselves, but appear to increase student motivation, which in turn will increase engagement and learning indirectly.

Katai and Toth note that applying games in learning, particularly to abstract concepts, might be more successful when viewed as providing multisensory learning opportunities, rather than teaching a singular skill in a particular learning style (2010). We discuss four papers that use games in their teaching; all of which use a single game, and three do so for a single assignment. Some papers have used existing games to teach concepts (Ng, Wang, Ng, & Loo, 2018; Rattadilok, Roadknight, & Li, 2018), some have designed new games (Lawrence, 2004), and some have used game development platforms (Overmars, 2004). These papers are not intended to be an exhaustive survey of work, but as a description of how games are typically implemented when teaching programming concepts.

Many projects have adapted existing games for teaching. Ng et al. used Minecraft to teach older students, in two different geographic locations, how to design, develop, and market an activity for young students (2018). The students were in two teams, one from Melbourne and the other in Kuala Lumpur, and were instructed to work in Minecraft. Students decided to create a build battle, where participants would be given a time limit and topic to build on, and they would create an object that could then be 3D printed. The study found that students did well at self-organizing and communication through email and messaging services, and that they were adept at using a variety of online resources to solve their own problems. Issues emerged when the students realized that many of their assumptions were not correct when designing for younger participants, and they learned to compromise and adapt their choices in the moment. The blended learning approach of working both online with others to design an activity and in-person with young participants whom they could observe and interact with, provided a variety of learning experiences that translate to real world applications. However, this was a singular activity that does not illustrate how learning was sustained over time.

Rattadilok et al. used an online mobile game, Clash of Clans (CoC), to teach machine learning concepts. Gamification was used specifically to attract students who might not be interested in computing topics otherwise, and to teach through an active learning approach (2018). Students used iGaME (In class Gamified Machine Learning Environment) to "develop a gaming strategy by applying a selected data set to the CoC" (p. 3). Students were able to track how their strategy was working by tracking the gameplay using their data sets. Students were able to take a more personalized approach to their learning by managing their own data sets, which expanded classroom flexibility. This process also made the student's progress visible to the instructor, so they could tailor their teaching to the student. iGaME is also able to generate new data sets from existing ones, which supports extended learning as students play through more levels of the game.

Some projects have designed and implemented unique games to teach their content. Overmars uses GameMaker, a high-level, drag-and-drop game creation tool to teach programming concepts (2004). Although GameMaker is understood as an "easier" tool, useful for building games without needing procedural coding, it still uses object-oriented, event-driven programming concepts. Overmars found that teaching students to design and implement games in GameMaker requires them to understand the relationships between objects, instances, properties of instances, and inheritance. Once students learn to design and build a basic game, their motivation tends to shift toward more advanced concepts such as behaviors, AI, or multiplayer options. Approaching programming in this way may seem too basic to some audiences, but it makes programming accessible to many populations and can support deeper knowledge of standard programming concepts.

Lawrence found that motivating students was an important part of teaching data structures, and used a game to support student engagement (2004). He developed a method called competitive programming to increase engagement, which was paired with game development to teach data structures. Students wrote code to develop the intelligence functionality for a board game, to determine the moves that their program would make. Students were evaluated on whether their code plays the game successfully and how well it does so. Student feedback indicated that this approach to teaching increased their interest in the course, made the course more interesting, and helped them become better programmers. The inclusion of a tournament feature appeared to be the most motivating, and student comments suggested that this competitive method caused them to invest more time and focus on their learning. Although this game was used for a single assignment and not over the course of a whole semester, it illustrates increased student motivation compared to other methods of teaching similar content.

LEARNING ENVIRONMENT EXPLORATIONS

Our class in Introductory Web Design has historically been taught in a variety of platforms, including as an active face-to-face class with lecture component, as an active face-to-face lecture class with online lectures for reference, as an online class, and as a blended class that includes a focus on games. The lecture-based platforms included lectures moving at a single tempo that students with a variety of skills needed to adapt to, but video and online resources facilitated students learning at different levels. Video or online-based classes allowed students to work at their own pace; however there was isolation from the rest of the class and it was difficult to address individual problems. Working face-to-face helped develop a shared understanding of successful design work and aesthetics, as well as shared problem-solving options.

By combining these options, we applied best practices in multiple ways, which allowed for a variety of learning paths and resources to support different learning paces, for social reinforcement, and introduced a collective aesthetic. Adding games to this approach increased engagement by exploring the concepts of code through new perspectives. Sedentary work at the computer can support isolation and hinder engagement even in a face-to-face classroom; playing physical and social games engages students in their space, with each other, and in the moment, for longer periods of time.

We found that creating a blended class where material is presented live, with the students actively coding alongside the instructor, having video lectures that they can refer back to, and then reinforcing this information with class exercises, games, quizzes, and problem-solving sessions has been a successful teaching strategy.

Motivation

Our course developed into a blended format over time, and games were added through this evolution to address student fears about starting to program. The course was taught online one semester, through which video lecture content was created. This content was kept available to students during face-to-face classes, which provided additional resources if the students needed to revisit not just the content directly, but to also see information leading up to certain content. Watching the programming process was valuable to the students: they benefitted from seeing the logic that goes into selecting what to code next, and also how to problem solve. This attention to the process (not just having clear cut solutions that could be memorized) has been valuable for students to review multiple times on the video content

if they don't understand it the first time in class. Programming is often learned by doing, and students seem to benefit the most when they are coding alongside the teacher and can see the results of their actions immediately (and ask further questions or to repeat the process).

Students often became more comfortable with coding once they started doing it, and the experiential approach helps students feel empowered through seeing their own successes. We wanted to include additional approaches, to make this jump more comfortable for students, and to support the approach to abstract concepts. We received a Teaching Innovation Grant from our Center for Teaching, Learning, and Technology to support the development of games that would target specific types of content and enhance student motivation and confidence. Students struggled initially with the high-level concepts of code. These include concepts of file paths (where is a file on your computer? On a server?), the basics of syntax, how code is organized spatially, and how code choices affect the spatial layout. Students do eventually grasp these concepts after trial and error explorations and applying them to projects, but we thought that exploring them in an additional medium might help them grasp concepts more quickly. Additionally, exploring coding concepts through games that use a variety of modalities (face-to-face, online videos, online resources), adds another facet of blended learning. We explored these concepts over two years (four semesters) in a face-to-face course, taught in two sections (for a total of eight different classes, roughly 120 students). We found that many of the objectives of the games aligned with current assignments in ways that we didn't expect: achievable goals, scaffolded activities, appropriate rewards, appropriate challenge, and intrinsic motivation.

Game Design Choices

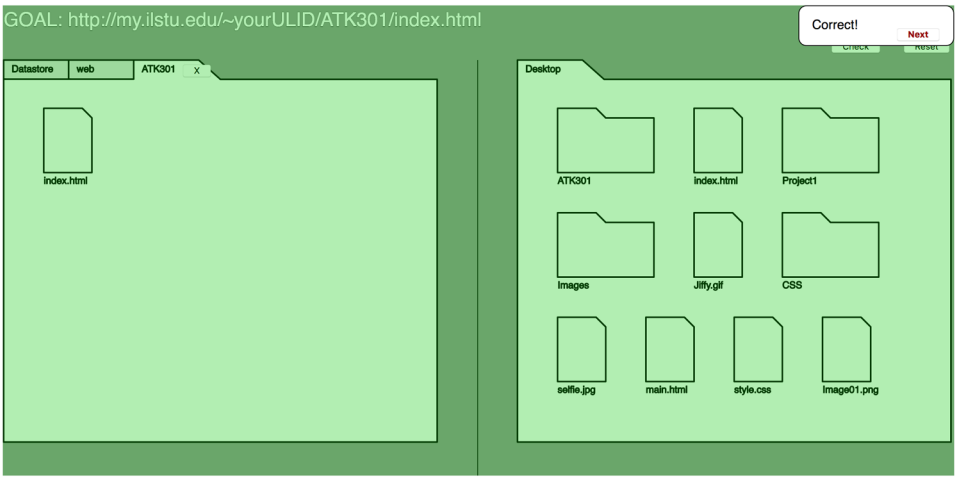
We began by brainstorming a variety of game concepts, and researched existing games for teaching code. We also explored tools for building games that were not as time-intensive as coding something from scratch. We narrowed down the game concepts to relate directly to the concepts being taught (avoiding superfluous game information). For example, one game concept was to use the game Operation as a metaphor for learning file paths. The interface would mimic a variety of files and folders, and students would need to find the file, select it, and move it to the correct folder/location without triggering the buzzer. However, we decided that this example for a game would rely too much on alternative knowledge and not represent the learning goals strongly enough.

We ended up developing, or finding and implementing, eight games to support specific topics in learning code. We describe these games with the modalities used, and how they supported a blended learning approach. We also use Kapp's discussion of learning domains in relation to gamification to further describe the type of learning that is taking place in each game (2012).

Pathfinders: Designed and Developed

This game was specifically designed to help students understand file management using an internal Illinois State University server system, particularly the differences between their local file system and the cloud-based system (see Figure 1). The game presented a file path example that required students to follow by placing files and folders in the right location. For example: Desktop/WebDesign/Images/cat.png. Students played alone, on their own computers, to complete all the file path challenges. The Pathfinders game was developed in Javascript using P5.js and was deployed on an independent server. Students could access the game through a web address. This game required students to understand the

Figure 1. Pathfinder game



file system on the own computers, and align their understanding of file paths with those online when using a server. By better understanding the concept of individual files, folders, local and online locations, and nesting, students developed a stronger understanding of their own machines and could apply deeper knowledge to the standard process of saving a file and being able to locate it. The Pathfinders game was an online modality which represented the use of space for filing away and retrieving files, and supported an active approach to file structures, and it focused on supporting declarative knowledge by sorting information. Although the students did not need to understand why they were doing this to be successful at the game, we discussed how the game reflected each student's organizational structure on their computer, and how important it was to be able to sort, place, and find files. This game supported blended learning by illustrating the importance of file paths to organize content on a computer, and how paths work, both locally on the student's computer and online. This game is available online for students to play at any time, especially to review content later in the course.

HTML Simon Says: Designed and Developed

We found that when students were just beginning to code, they had a difficult time remembering what items needed to be at the top of the page when coding, and what items needed to be at the bottom. Because these spatial concepts of information are similar across many different coding languages, we felt it was important to find alternative ways to reinforce this knowledge. We developed a game of Simon Says that students could play in real, physical space to understand basic organization concepts of HTML (see Figure 2). HTML is a script-based language that, like other languages, initializes settings information, parameters, and libraries in the top section of code, and builds the visible parts of a webpage in the bottom section of code. By using large-font printouts of code, we placed segments of code in spatial organization within a large room. Students were organized into two groups to play against each other. We used a variety of exercises to identify pieces of code and their location, to construct, and deconstruct pieces of code to achieve different outcomes. For example, we began with questions such as:

Figure 2. Simon Says game

- "Where is a link to an external page?"
- "What code indicates the cultural language we are using?"
- "What tag would indicate a subheading on a page?"

This game was designed to support a spatial understanding of code structure. Large physical cards with code segments printed on them were arranged on the floor to create various programs. Students worked as groups in timed exercises to create correct and successful code. Students began with an open period to construct the correct code, then the printouts were shuffled and they had a timed race to complete the code again. Later in the class, we provided a wireframe, a low fidelity sketch of a webpage layout with title, central navigation, image, and paragraph text that students needed to construct using the printouts that they had. Because the use of space in programming is conceptual, we used physical space as a metaphoric device to support sorting and matching information. This game does not support blended learning directly, but does support multisensory learning. By creating social structures and competition in the course students also get to know each other better, and learn who is a good reference to ask for help later on.

HTML Super Markup Man: Existing Game

In order to continue working on their understanding of syntax, we searched for existing games that could support student's knowledge. Through our research we found a pre-existing 2D game that required players to move bits of code around to create correct syntax. Students played on their own to complete all the challenges by making the right combinations of code and punctuation. Although the game is online, we found that students worked together on particularly challenging levels, especially when they couldn't see their own errors, such as spaces in the code. This game focused on supporting declarative knowledge by sorting, matching, and association of information. For example, students only have so many pieces of

information to make the right syntax, so they used a combination of deduction and comparison to syntax in prior levels. The game is online and can be played anytime, as well as referred back to. Students have noted that they went out looking for other games online to explore HTML and CSS after this game. This game can be found online at: http://markup.roppychop.com/

Exquisite Corpse Interactive Stories: Designed and Developed

Exquisite Corpse is a community-based art strategy where someone starts creating in a particular platform (for example, painting a landscape), and person take turns adding content. Each person's individual content should be thematic (i.e., a student would develop one concept, not three, to add to the painting) and is consecutively added to the work. Like a game of telephone, the original concept will morph and change depending on what is added.

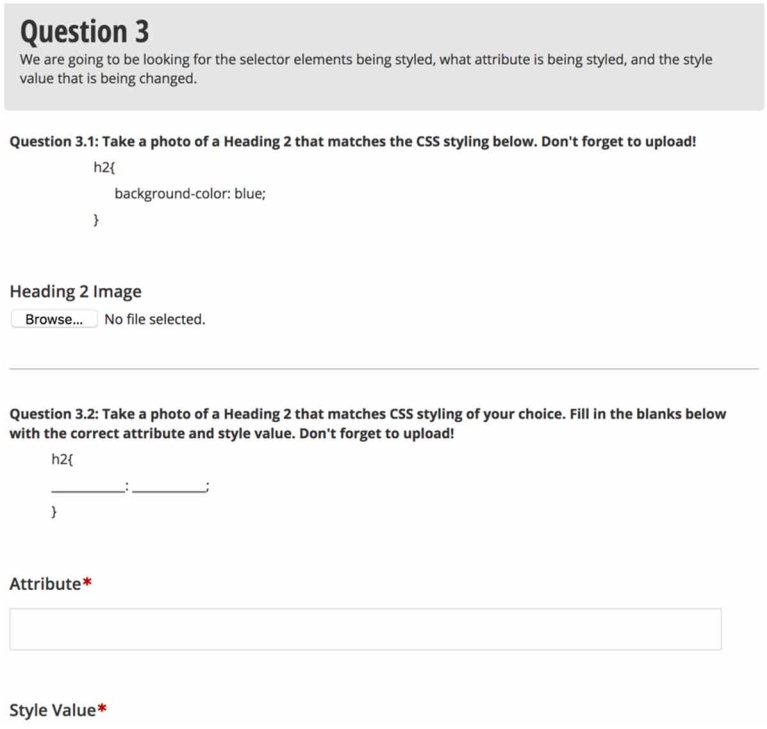
In our classes we used Exquisite Corpse as an activity to sequentially build an interactive story, where each student created a phrase, image, movie, audio clip, or interaction. Students indicated what they added, and where it should go; such as a location on an existing page, or to add a new page. The resulting story is always silly, yet illustrates concepts of page structure and navigation (for example: a squirrel that misread the groundhog's Spring prediction and is confused enough to search for Spring by launching themselves into outer space). Through this method, students collectively worked on creative concepts, and made procedural choices about code. Although the activity occurs on the teacher station's computer, students actively discussed options as a class. If an image is required, the teacher searched images on the web and we collectively found resources that fit the story. This activity provided another active method for students to experience the concept and synthesize their knowledge in another creative medium, beyond creating pragmatic web pages from a top-down structure. This game specifically supported blended learning by encouraging students to look up content online while writing code live and constructing a story in front of the class. Students were allowed to help each other out to figure out the code that was needed to achieve their vision.

CSS Scavenger Hunt: Designed and Developed

HTML code refers to the structural components of web development, and CSS refers to the stylistic aspects. CSS uses a completely different syntax (code structure) from HTML and is often confused and mashed together by students. Though there are ways to integrate the two languages elegantly, beginning students usually do not understand enough about each language to make a functional project. Because CSS is working with stylistic components, properties such as text color, fonts, and line-weight can be specific targeted in code. To better understand how to "see" these code functions, we created a scavenger hunt game for students could go out in the world to find these functions (see Figure 3).

Students were required to pair up to search the campus for a variety of style components. The purpose of going out into the world was to see their code attributes "in the wild," finding examples of styles on poster boards and flyers. Although this game could have been played by searching options online, after the success of Simon Says as a spatial game, we felt that getting students physically outside could be beneficial for seeing how styles manifest themselves all around us. The game was designed in FormStack, our local university form tool, so students could both answer questions as well as include images and descriptions of the found style as evidence. An example scenario is that students would get an email with one of two forms. There were two versions so that partners could work independently but still

Figure 3. CSS scavenger hunt game



together, so each partner would choose either Form 1 or Form 2. By using their smartphones to access the form and search for online resources, and going around campus to search for physical applications of style, students applied their learning in multiple modalities. This helped students connect their more abstract understanding of code to works in a specific environment to the world around them. This game supported conceptual knowledge through attribute classification, where students had to search for and document attributes that fit their problem. This game also supported blended learning by first requiring students to test their knowledge, but could search online for additional support or instruction as needed. Because students were working in pairs they have noted that they learned how to better problem solve and search for the right keywords from their partner during this activity.

CSS Sushi Diner: Existing Game

This pre-existing game investigates hierarchy in CSS, while helping students attend to syntax to solidify their knowledge of CSS. By using the concept of sushi as a metaphor, the player targets various items on a plate. Students play on their own to complete all the challenges, with hints appearing on the right pane. Although the game is online, we found that students would work together on particularly challenging levels, especially when they couldn't see their errors, such as which item they were targeting. Similar to the HTML Super Markup Man game, this game focuses on supporting declarative knowledge by sorting, matching, and associating information. The game is online and can be played any time, as well as referred back to. This game can be found online at: https://flukeout.github.io/

Jeopardy: Existing Game

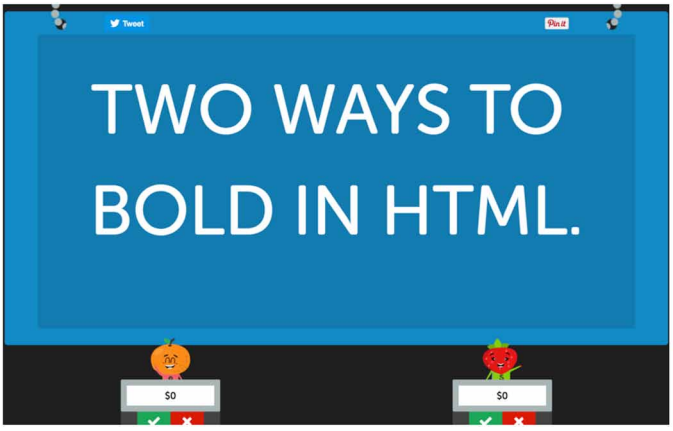
By the mid-point in the semester, students should have an understanding of HTML and CSS code that can be applied to developing a basic website. To test everyone's knowledge and remind students of content from earlier in the semester, we created a Jeopardy game (see Figure 4). We used an existing game platform, Jeopardy Rocks, to develop a game using two teams to answer questions that reviewed HTML and CSS syntax. Although students were given the option to use web resources to answer questions, the response period was timed and it was often quicker for them to get help from a teammate rather than go researching. We played one incomplete version without the timer as a refresher so that students could remember the content in this social scenario (and off the page of their coding platform), and then played a complete version with the timer. By bringing content review into a social platform, students had to understand their programming knowledge differently which helps to solidify their engagement with it. Playing Jeopardy reinforced declarative knowledge by repeating information in a social environment with rules and competition. This game specifically supported blended learning by encouraging students to look up content to help out their group. Students also noted that there were some answers they didn't believe were correct so they went back later to check and re-examine on their own. Level 1 of this game can be found online at: https://www.jeopardy.rocks/webterminology

Level 2 of this game can be found online at: https://www.jeopardy.rocks/beginningcss

METHODOLOGY

This chapter does not focus directly on student performance when learning through playing games, but rather on the reported engagement about what games were fun, what they learned the most from, and what they still struggle with (even after playing games). The class structure was that students learned specific concepts through follow-along lectures and in class activities on each topic for 1–3 weeks, depending on the complexity of the topic. Games were then introduced as reinforcement, or to provide a new perspective on the topic once the basics were acquired through the blended approach of live lectures, activities, and video tutorials. Assignments were broken down to address small components of informa-

Figure 4. HTML/ CSS Jeopardy Game



tion, and were mostly given before games were played (to build up the larger pool of knowledge). A large assignment was then given after games were played, to reinforce information before applying it in more complex ways. The mix of game platforms (live/in-person such as Simon Says, online such as Super Markup Man, or mixed such as Jeopardy where students could search the answers as needed) supported the mix of student's current level of knowledge. Association or deductive methods could be used in some situations to find the answer in the moment without support, or learning to apply the right keywords to search for an answer online could be used in other situations. However, online approaches to problem-solving are sometimes challenging. Although code syntax does not often change, the frameworks used (e.g., Macromedia Flash) become outdated, and so the answer that will work for the problem might not be first in a keyword search result. We believe that students learned to use both approaches in problem-solving in order to know what approaches will most likely work (and what keywords to use to search for online support), but do not have direct evidence of this beyond some anecdotal student reporting.

We gathered data on students' engagement and learning in gameplay in two stages: during gameplay, through instructor observation and immediate student reporting, and at the end of term with surveys about the games in the course. Data was collected over four semesters. Students' enjoyment and engagement with the games was documented by observing and taking notes on the students' commitment to playing, teamwork, and conversations with peers during gameplay. We followed-up in the course right after the gameplay to discuss if the students found the games helpful to understanding the concepts. We also provided surveys at the end of the term to ask students what activities and assignments were the most fun, the most challenging, which they learned the most from, and what other ideas for games or assignments would support their learning. These three timelines of qualitative data gave us feedback about what games were working, how they were working, and what else students were wishing for or could use in future iterations.

OBSERVATION AND SURVEY RESULTS

General Instructor Observations

We observed student interactions through games over four semesters, and took notes about general student engagement. The analysis of these notes showed trends across the semesters. Students that were already comfortable with the course material engaged easily with the games and seemed to find them enjoyable to practice their skills. However, students that were less comfortable with the course material found that they needed to engage more with their peers to complete the games, but did not find this discouraging. When working on solo games these students were able to learn through the gameplay or source information on their own to complete the challenges. These students also seemed to be able to search for their own resources more easily later in the semester, which concurs with studies that show positive impacts of students using technology in engaged manners to support self-directed learning (Rashid and Asghar, 2016). Students that were struggling with the material sometimes felt embarrassed to be in a group setting and would sit out or disengage. However, if they were working on a solo game they would often persevere and attempt to grasp the coding concepts through the game. These peer to peer interactions also continued throughout the semester to share problem-solving methods and resources, which helped empower students to troubleshoot themselves. The focus on games did not always seem helpful as a starting point for coding content, but they worked well as a reinforcement option.

In discussion after gameplay, the students mostly echoed our observations. Students who were able to use the games as reinforcement to their learning or a helpful challenge to source the needed information were very positive about having a playful component to the course. Some students who grasped the information very easily found the games to be boring, and were ready to move onto new projects. This situation could be evidence of expertise reversal effect, where certain learning experiences no longer work for learners at a certain level of expertise, such as a more experienced learner not getting anything out of an assignment the same way a novice does (Chen, Kalyuga, & Sweller, 2017). However, students who struggled with learning the material recognized that they needed to catch up and prepare a bit more before playing the games. Students also noted the usefulness of practicing a skill in different formats, such as from lecture by coding along, to playing by moving things around spatially, to creating a silly webpage with their peers. Although every assignment and game were not considered successful by every student, every time, there appeared to be consensus that exploring material through a variety of different methods supports learning.

Post-Course Surveys

We had 51 students participate in post-course surveys over four semesters. The games and assignments stayed the same for these four semesters, but the way the survey questions were worded differed slightly between the first semester, the next two semesters, and the last semester. These changes will be discussed as Stage 1, 2, and 3. In Stage 1, questions asked about what activity was hardest, but the responses did not include much information on the games (which suggests that the students did not see them as part of class activities, but as something extra). In Stage 2, we asked questions designed to encourage game names explicitly, as part of class activities. Because we had such a strong response to both traditional assignments and games in this survey, we separated them for the final stage, asking in Stage 3 "What was the hardest game?" and "What was the hardest activity/ assignment?". Demographic information on the students was not recorded for this study in an effort to keep responses fully anonymous. Student performance was also not included in this chapter due to issues in confounding variables between different instructors and assignments.

Post-course surveys included the following questions in Stage 1:

What was the most fun activity in class this term?
What activity/ assignment did you learn the most from?
What was the hardest assignment/ activity? Why?
What is a topic that you struggle with still?
If you could learn through a game or teaching someone else, what class topic(s) would be best suited?
If you could play a game to learn a class topic, what would you design? Pick a topic and design a game for it.

Post-course surveys included the following questions in Stage 2:

Did you feel that playing games helped you learn to code? If so, how?
What was the most fun game/ activity/ assignment in class this term?
What game/ activity/ assignment did you learn the most from?

Table 1. Stage 1 Student Responses

Most fun activity:	Final Portfolio Website: 5	31%	Simon Says: 2	12.5%	Rest: Singular Options
LA:	Experience the Concept		Conceptual Metaphor		
Activity learned most from:	Recreate a Movie Poster: 6	37.5%	Final Portfolio Website: 5	31%	Rest: Singular Options
LA:	Association/ Declarative		Experience the Concept		
Hardest activity:	First HTML/ CSS Combination Assignment: 6	37.5%	Final Portfolio Website: 2	12.5%	Rest: Singular Options
LA:	Association/ Sorting		Experience the Concept		
Topic student still struggles with at end of term:	Navigation		Spacing/ Margins/ Floats		Javascript/ JQuery

What was the hardest game/ activity/ assignment? Why?
What is a topic that you struggle with still?
If you could design a game to teach someone else, what class topic(s) would be best suited?
If you could design a game to learn a class topic, what would you design? Pick a topic and design a game for it.

Post-course surveys included the following questions in Stage 3:

What was the most fun activity in class this term?
What was the most fun game that we played in class?
What activity/ assignment did you learn the most from?
What game did you learn the most from?
What was the hardest assignment/ activity? Why?
What was the hardest game? Why?
What is a topic that you struggle with still?
If you could learn through a game or teaching someone else, what class topic(s) would be best suited?
If you could play a game to learn a class topic, what would you design? Pick a topic and design a game for it.

We analyzed the data by stage, based on the topics from the questions of: fun, learning, hardest, other topic to design for. We focus on these topics to assess how the students perceive the use of games amidst the other assignments in their class, and we note the areas that they would like to have additional playful support. Typically, each question has two to three top answers which we note here. Learning approaches (LA), as noted in Kapp, are included to compare types of knowledge as a reference (2012).

Stage 1: One Semester: 16 Responses

Stage 1 was the first semester that we introduced the games to the students, and were still developing games as we went through the semester. Because this seemed to be a big event that was transparent to the

students it is surprising that so few of the responses noted the games played. We attributed that to asking all the questions around "activities," and assuming that students would automatically see the games as an integral part of the semester's assignments (especially because playing the games was implemented into assignment grading, students had to participate to get a mark). We include select student comments below to illustrate the qualitative information we received, and contextualize it with the quantitative information.

Comments addressing the most fun activities include:

I like looking online for inspiration when creating my own website design.

Not sure. All of it to me was fun, I really love code!

The Final Portfolio Website assignment was noted as the most fun activity at 31%, with Simon Says as the next fun activity at 12.5%, with the rest of the responses as singular options. In the comments, students mention looking online for inspiration, suggesting that they were able to look for materials outside of the course to work with. This would reflect on the Final Portfolio Website assignment, where students could create their own designs, although they also needed to figure out how to implement them. This implementation often required them to learn new techniques on their own. While games are not specifically mentioned, students enjoyed learning to code, suggesting that the course as a whole was successful at sparking an interest in programming.

Comments addressing what activities students learned the most in:

I learned the most from designing the poster because there was a lot of different ways to figure it out.

Recreating a movie poster was noted as the assignment students learned the most from at 37.5%, with the Final Portfolio Website second at 31%. This comment targets the poster assignment where students were recreating a visual image of a movie poster with code, by using spacing concepts and targeting specific elements of code with classes and IDs.

Comments addressing the hardest activities include:

The final portfolio website. Because it challenged everything that I learned this semester.

The first HTML/ CSS combination assignment is noted as the hardest at 37.5%, with the final portfolio website at 12.5%. This comment states that the final assignment to use all their knowledge, compiled over the semester, was the hardest by bringing everything together in a different way. Although this comment does not reflect the quantitative data, it illustrates a perspective to how an assignment was challenging, by requiring the student to use everything they learned to be successful.

Comments addressing topics the student is still struggling with include:

Properly spacing out different parts of a website.

Javascript

These comments suggest that two of the typically more challenging topics, managing space and learning Javascript, were still issues at the end of the course. Quantitative information notes navigation, spacing, and Javascript as challenges, which aligns with the qualitative feedback.

One additional comment that was not included in the selected questions but related to them is about coding in general:

The topic question was: If you could play a game to learn a class topic, what would you design? Pick a topic and design a game for it.

I liked the game we played earlier in the semester with the coding. That helped a lot by

understanding how to look at something and then transforming it into code. I would go

somewhere along that same concept to design a game.

This comment suggests that (probably the Simon Says game) was useful for the student to understand a spatial structure in code and have a variety of opportunities to manipulate it, and directly notes a metaphoric concept as being helpful with comprehension (Kapp, 2012). This note, correlated with high quantitative evidence through the three stages of data collection (Stage 1: #2 most fun activity, Stage 2: #2 most fun and #1 learned most from, Stage 3: #1 most fun activity, #1 most fun game, #2 learned most from) that a field game was useful to understand code (which has not come up in any other literature) suggests that this approach could be used for other concepts.

The traditional assignments that do come up regularly as most fun, most learned through, and hardest activities include the final project to develop their own personal portfolio website (experiencing the concept focus), the first assignment in which they combined HTML and CSS together (association/declarative knowledge focus), and a project where they recreate a movie poster in code (association/declarative knowledge focus). Although playing Simon Says (metaphoric concept) and Jeopardy (reinforcement focus) do come into the results, it is these three prior assignments that show up regularly and appear to be the major points in the semester where students are both challenged and making the most progress. Students noted continued struggles with topics of navigation, spacing, and JQuery which are some of the more complex topics, which are more conceptual and problem-solving focused concepts, and less declarative.

Stage 2: Two Semesters: 15 Responses

In Stage 2 of the surveys we were concerned that there were not more responses about the games, and so we specifically added the word "games" to the questions in case students were not considering them part of class activities. In these two semesters we ended up with results that mostly focused on the games, so are quite different from the responses we got in Stage 1. The 2D Platformer game Super Markup Man was reported as the most fun, and Simon Says closely followed. Simon Says was also noted as the game most learned from, with Sushi CSS game ranking just behind. The Sushi game was also noted as the hardest game, with the development of the Final Portfolio Website noted as well. Students continue to note a struggle with topics on spacing and JQuery, the more complex topics to manage, which is not surprising. Suggestions for new games reflect these topics with students asking for a new game around

Table 2. Stage 2 Student Responses

Did games help coding?	Yes: 13	87%	No: 2	13%	Maybe: 1	7%
Most fun game/activity/ assignment:	Super Markup Game: 7	47%	Simon Says game: 5	33%	Sushi Game: 2	13%
LA:	Sorting/ declarative		Metaphoric concept		Sorting/ declarative	
Game/ activity/ assignment learned most from:	Simon Says game: 6	40%	Sushi Game: 4	27%	Super Markup Game: 2	13%
LA:	Metaphoric concept		Sorting/ declarative		Sorting/ declarative	
Hardest game/ activity/ assignment:	Sushi Game: 10	67%	Final Portfolio Website: 2	13%		
LA:	Sorting/ declarative		Experiencing the Concept			
Topic student still struggles with at end of term:	Spacing/ Margins/ Floats		Javascript/ JQuery			

spacing in particular that could be modeled after Simon Says and be a field game so they could physically understand the concepts of moving code around.

Comments addressing whether games helped the students understand code:

I didn't play enough of the games for them to teach me code.

There were some games that aided me more than others. Overall it was a nice alternative to a constant lecture.

At first, I didn't think that games would apply to learning but now I see different ;) !

Not really. I didn't enjoy playing them. They just made coding seem more frustrating.

I think it was a good way to physically visualize what coding was and how it was structured.

This selection of comments shows the breath of student experiences, while the numbers suggest that students did enjoy this method (87%) compared to students who did not enjoy this method (13%) and students who stated "maybe" (7%). The comments note that some students did not enjoy the games, or didn't do enough to impact them. Some may have helped, while others didn't. While some people found them at least a nice change, other did find them a useful method for understanding code.

Comments addressing the most fun game/activity/assignment include:

I don't remember what it was called, but I liked the game with the person jumping from line to line, picking code in order. It was easy and simple but it made me feel like I was really learning a lot, which I have!

The most fun and interesting one was when we were rearranging code on the floor against another team.

The first comment refers to the Super Markup Man game, where students pieced together code, which has a sorting/declarative knowledge focus and had a 47% positive response. The second comment refers to Simon Says, which uses metaphoric concepts and also mentions working against another team. Simon Says was the second most fun game in this survey, with a score of 33%. These comments suggest the game designs had achievable challenges, both by themselves and in groups, and were supportive to the learning process.

Comments addressing the game/activity/assignment where most was learned include:

The file/folder game we played like the second week of class. It was a good refresher for how to structure files so that everything shows up correctly on your website.

The Pathfinder game supported student's understanding of file paths when working with servers, which is different from their local laptop computer. This was helpful to see that it was considered a successful tool that supported their choices for the rest of the term, even though it was not noted in the quantitative data (Simon Says game: 40%, Sushi CSS game: 27%, Super Markup game: 13%).

Comments addressing the hardest game/ activity/ assignment includes:

The Sushi game; I don't remember there being a tutorial so you had to kind of figure it out on your own, and there was no text so there were no hints for what exactly it wanted to let you to pass the level.

The Sushi CSS game was set up to have hints on the right side of the screen, but they seemed a bit cryptic until you figured out how to use them. It took many students a while to understand the hints and the topic was challenging, dealing with hierarchy and inheritance which we don't cover deeply in class. While the Sushi game relied heavily on association, it was in a form that took a while to figure out (and once the students did figure it out they could go much further). However, figuring out the game structure often turned off students.

Stage 3: One Semester: 15 Responses

In Stage 3 we changed the survey language again, specifically to separate the concept of activities/assignments and games. While using the word "game" did give us responses relating to specific games, the term "activities" provoked responses including both assignments and games, suggesting that a shift to "assignments" and "games" might help create a separation. However, at this point games were an embedded component of the course and students may have been seeing them as just another assignment, that is not a separate concept in the course compared to Stage 1 of the project.

The most fun activity question highlighted three games, Simon Says (40%) and the CSS Scavenger Hunt (27%) alongside the Exquisite Corpse (13%) activity. Responses for the most fun games and games most-learned-from included: Super Markup Man, Simon Says, and Jeopardy. The activities most-learned-from and the hardest activities are also very close in student response, reflecting the CSS replicate assignment, recreating a movie poster, and the final portfolio assignment. In Stage 3, the hardest games were noted as Super Markup Man and Jeopardy. It is interesting that the Sushi game was not noted much in this stage after the strong response in Stage 2.

Comments addressing the most fun game include:

Table 3. Stage 3 Student Responses

Most fun activity:	Simon Says game: 6	40%	CSS Scavenger Hunt: 4	27%	Exquisite Corpse Website: 2	13%
LA:	Metaphoric Concept		Attribute classification/ conceptual		Experience the concept	
Most fun game:	Simon Says game: 7	47%	Super Markup Game: 6	40%	Jeopardy Game: 4	27%
	Metaphoric Concept		Sorting/ declarative		Repeating/ declarative	
Activity learned most:	CSS Replicate Assignment: 5	33%	Recreate a Movie Poster: 4	27%	Final Portfolio Website: 3	33%
	Sorting/ declarative		Association/ Declarative		Experience the concept	
Game learned most:	Super Markup Game: 9	60%	Simon Says game: 5	33%	Jeopardy Game: 2	13%
	Sorting/ declarative		Metaphoric Concept		Repeating/ declarative	
Hardest activity:	CSS Replicate Assignment: 8	53%	Final Portfolio Website: 6	40%	Recreate a Movie Poster: 2	13%
	Sorting/ declarative		Experience the concept		Association/ Declarative	
Hardest game:	Super Markup Game: 7	47%	Jeopardy Game: 6	40%		
	Sorting/ declarative		Repeating/ declarative			

When we walked around and took pictures of CSS styles around the building.

I got to stand up and move around and it was competitive.

In these comments, students note "moving around" which suggests that physically doing something was fun, and possibly the multisensory component was helpful (Katai and Toth, 2010). Competition is also noted as being interesting. Quantitative data shows that Simon Says (another metaphoric concept game is noted as #1 at 47%, the Super Markup Man is #2 at 40%, and Jeopardy at 27%).

Comments addressing the activity/assignment where students learned the most include:

The assignment I learned the most from was the one where we were given a goal image.

The Team Coding exercise we did in the CVA.

These comments note the assignments (Poster Recreation–33%– and CSS Replication–27%) where they were given a goal image to recreate, which is a specific assignment with clear tasks and challenges. These assignments were scaffolded to use specific knowledge that had just been taught in the classes. The team coding exercise refers to the Simon Says game, which again suggests the field game approach was helpful.

Comments addressing the hardest activity/ assignment include:

Final Project; We have to make our designs come to life and actually work! And it can be tedious sometimes, but it is worth it in the end.

While this Final Portfolio Assignment ranked as #2 at 40%, it was called out both in Stage 1 and 2 as a challenge to put together all their knowledge from the semester into one assignment with few guidelines.

Comments addressing the hardest game include:

Not really any hard game, but Jeopardy really racked my brain.

Simon Says because it was under a time limit

The hardest game was the Roppychop [Super Markup Man] game because as you would progress in the game it kept getting more challenging and frustrating

These comments noted different games for different reasons: Jeopardy racked my brain, being under a time limit, as you progress...it gets more challenging. These comments note both cognitive and sensory challenges that were indicated as hard. Quantitative data reflects these comments with Super Markup Man at 47% and Jeopardy at 40%.

Overall Stage Findings

Regardless of our wording to bring out responses about games, activities, and assignments, trends were visible across all three stages. The Simon Says, Roppychop, Jeopardy, and Sushi CSS games were regularly noted as being fun, providing learning opportunities, and (especially the Sushi game) being difficult. More traditional assignments that students had fun with or learned a lot from were also often the difficult assignments requiring more conceptual knowledge and problem-solving: recreating the movie poster and the Final Portfolio Website. Concepts that regularly connect the game and traditional assignment activities are those of translating visual space, whether in the design or in the organization of code. We do wonder in hindsight how the language choices of "game," "activity," and "assignment" were perceived by the students—did they see a difference between game and activity? Are there generational differences in how "games" are viewed, or was the survey word-choice the biggest effect on their choices?

Most of the collaborative/social games were regularly mentioned. The Simon Says game was highly popular, which might be due to the direct competitive and social nature of game play. It is a game of movement, which is unusual for learning code, but seemed to help students make correlations between how they visually make spatial choices in design with the spatial choices they need to make when writing code (a movement approach to learning code was also highly successful for multisensory reasons in the work by Katai and Toth, 2012). The Jeopardy game also uses a direct competition in a social atmosphere, and though it was used only to reinforce knowledge that had already been developed, it too was very popular.

Roppychop, aka Super Markup Man, is a good example of a successful solo and declarative knowledge game. It's frustrating, challenging, and hard but the students also felt they learned from it. The Sushi CSS game was too difficult, and so many students did not enjoy playing it. Super Markup Man hit the right balance, like a good assignment does: it challenged but didn't defeat the student. For future work we would like to explore how to make the solo, individual games collaborative (e.g., including leaderboards, competing against other students, etc.).

DISCUSSION AND CONCLUSION

Even though we did not intend to see as many comments on the traditional class assignments, there are parallels to the games. Re-creation activities, such as the "build from an image" assignments, were popular and noted as activities that students learned a lot from, and that they understood what they needed to do. The more scaffolded, specific activities, such as the Super Markup Man game and recreating the movie poster assignment, were popular because they used specific steps, challenges, and levels to build off of existing skills and develop new skills along the way. Concepts of balance between challenge and rewards is important here: assignments that had challenges the students didn't feel ready for were considered too hard (such as the first assignment putting HTML and CSS together, and the Sushi CSS game). Activities where students had to create their own challenges, such as a sandbox-type game, were more stressful but equally more rewarding (such as the Final Portfolio Website). These approaches to game design that reflect learning outcomes are mirrored in Kapp's work (2012).

Translating visual space into code structures by playing Simon Says as a field game was regularly noted as helping students better understand why they were doing what they were doing. Walking places or moving around was noted in many of the comments, such as the CSS Scavenger Hunt and Simon Says, which suggests that getting away from their computers was helpful to provide new perspectives. This suggests that research in multisensory learning, such as by Katai and Toth, is valuable to provide students with many approaches to their learning in terms of modality (2010).

This use of variety is also useful when considering games and activities played solo vs. in a group. Students playing a game on their own were able to focus on their own challenges and would persevere even if they got stuck, whereas in a group game some might defer to others. Yet, sharing the problem-solving process with others helped them see how others approach a problem and could then look for other resources or approaches in their own problem-solving processes.

The gameful approach seemed to be most positive with students who were already very comfortable with the material and could easily engage. Obviously, students who were behind were not as comfortable playing the games (especially group games) because it was obvious they could not be successful. Students who reported that they understood the concepts well noted that they were bored, but these comments are also tied to group scenarios, so they may not have been interested in the social aspect. Yet, all student feedback suggests that games with a variety of modalities, a variety of group or solo scenarios, a variety of specific and scaffolded activities as well as open and ill-defined activities were useful if they could find that sweet spot that balances challenge and rewards.

Some papers discuss the use of games in teaching as an automatic approach to blended learning (Lawrence, 2004; Ng et al., 2018; Overmars, 2004; Rattadilok et al., 2018). While the use of games does diversify the teaching approaches and can provide a technological element to an otherwise in-person class, this description of blended learning does not seem fully synonymous with that in the education literature. Blended learning in the education literature describes a combination of online/ technological approaches with face-to-face instruction (Bourne and Seaman, 2005), which relies heavily on thoughtfully crafted learning environments to provide more engaging learning experiences (Garrison and Vaughan, 2008; Katai and Toth, 2010). In this chapter we leverage a variety of games that shift into the technological domain, and also come back to face-to-face and live social environments. When games are combined with assignments that scaffold learning from declarative knowledge into more complex conceptual knowledge there are benefits in practicing and solidifying both technical, skill-based information as well as more open-ended and complex designing and problem-solving situations. Karl Kapp notes that

many elements that make successful games also make successful learning experiences: achievable goals, clear feedback, control over actions, scaffolded activities, appropriate rewards, appropriate challenge, and intrinsic motivation to name a few (2012, p. 74).

Our blended learning approach required specific choices in the game selection and designs to combine methods of solo and social play, competition, online and live play, and especially motivation and achievable goals that would specifically be tailored to the information being taught. While some of these games are long-existing, or already designed to use typical methods to teach a course content (e.g. Super Markup Man and a side scroller level game), the multi-sensory choices used support the blended learning environment by allowing students (and possibly teaching them) to use a variety of resources such as each other, the internet, or their own problem-solving skills. Our courses in particular have students at many different levels, so having a variety of course content approaches is supportive to either scaffold their knowledge, motivate their learning, or reinforce what they already know. By combining the playful approaches to teaching within a blended learning environment, with meaningful choices created a structure that students can navigate at a variety of paces to develop their own problem-solving skills.

This chapter included a background on the use of games to teach programming and current approaches to blended learning, while focusing on the author's strategies, implementation, and evaluation of games in a blended classroom.

ACKNOWLEDGMENT

This research was supported by the Center for Teaching, Learning, and Technology at Illinois State University.

REFERENCES

Ak, O., & Kutlu, B. (2017). Comparing 2D and 3D game-based learning environments in terms of learning gains and student perceptions. *British Journal of Educational Technology*, *48*(1), 129–144. doi:10.1111/bjet.12346

Baepler, P., Walker, J. D., & Driessen, M. (2014). It's not about seat time: Blending, flipping, and efficiency in active learning classrooms. *Computers & Education*, *78*, 227–236. doi:10.1016/j.compedu.2014.06.006

Bellotti, F., Kapralos, B., Lee, K., Moreno-Ger, P., & Berta, R. (2013). Assessment in and of Serious Games: An overview. *Advances in Human-Computer Interaction*, *2013*, 1–11. doi:10.1155/2013/136864

Bonk, C. J., & Graham, C. R. (2006). *The handbook of blended learning: Global perspectives, local designs*. Hoboken, NJ: John Wiley & Sons.

Bourne, K., & Seaman, J. (2005). *Sloan-C special survey report: A look at blended learning*. Needham, MA: The Sloan Consortium.

Chen, O., Kalyuga, S., & Sweller, J. (2017). The expertise reversal effect is a variant of the more general element interactivity effect. *Educational Psychology Review*, *29*(2), 393–405. doi:10.100710648-016-9359-1

Cohen, D. (2007). *The development of play (Concepts in developmental psychology)*. Routledge.

de Freitas, S. (2018). Are games effective learning tools? A review of educational games. *Journal of Educational Technology & Society*, *21*(2), 74–84.

Dewey, J. (1916). *Democracy and education*. New York: Macmillan.

Dicheva, D., Dichev, C., Agre, G., & Angelova, G. (2015). Gamification in education: A systematic mapping study. *Journal of Educational Technology & Society*, *18*(3), 75–88.

Farber, M. (2018). *Game-based learning in action: How an expert affinity group teaches with games*. Peter Lang.

Gardner, H. (2000). *Intelligence reframed. Multiple intelligences for the 21st century*. New York: Basic Books.

Garrison, D. R., & Vaughan, N. D. (2008). *Blended learning in higher education: Framework, principles, and guidelines*. Hoboken, NJ: John Wiley & Sons.

Gee, J. P. (2003). What video games have to teach us about learning and literacy. *Computer Entertainment*, *1*(1), 20. doi:10.1145/950566.950595

Gee, J. P. (2007). *Good video games and good learning: Collected essays on video games, learning and literacy (New literacies and digital epistemologies)*. Peter Lang. doi:10.3726/978-1-4539-1162-4

Han, F., & Ellis, R. A. (2019). Identifying consistent patterns of quality learning discussions in blended learning. *The Internet and Higher Education*, *40*, 12–19. doi:10.1016/j.iheduc.2018.09.002

Kapp, K. (2012). *The gamification of learning and instruction: Game-based methods and strategies for training and education*. San Francisco, CA: Pfieffer Publishing.

Katai, Z., & Toth, L. (2010). Technologically and artistically enhanced multi-sensory computer-programming education. *Teaching and Teacher Education*, *26*(2), 244–251. doi:10.1016/j.tate.2009.04.012

Kolb, D. A., & Fry, R. E. (1974). *Toward an applied theory of experiential learning*. MIT Alfred P. Sloan.

Lawrence, R. (2004). Teaching data structures using competitive games. *IEEE Transactions on Education*, *47*(4), 459–466. doi:10.1109/TE.2004.825053

Malone, T. W. (1980). What makes things fun to learn? Heuristics for designing instructional computer games. *Proceedings of the 3rd ACM SIGSMALL Symposium and the First SIGPC Symposium on Small Systems*, 162–169. 10.1145/800088.802839

McCarthy, J., Wright, P., Wallace, J., & Dearden, A. (2006). The experience of enchantment in human–computer interaction. *Personal and Ubiquitous Computing*, *10*(6), 369–378. doi:10.100700779-005-0055-2

Navrat, P. (1994). Hierarchies of programming concepts: Abstraction, generality, and beyond. *ACM SIGCSE Bulletin*, *26*(3), 17–21. doi:10.1145/187387.187397

Ng, D. C., Wang, W. F., Ng, D. L., & Loo, Y. T. (2018). A blended learning approach to experiential STEM education for young learners. *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering*, 916–920. 10.1109/TALE.2018.8615302

Overmars, M. (2004). Teaching computer science through game design. *Computer*, *37*(4), 81–83. doi:10.1109/MC.2004.1297314

Philpot, T. A., Hall, R. H., Hubing, N., & Flori, R. E. (2005). Using games to teach statics calculation procedures: Application and assessment. *Computer Applications in Engineering Education*, *13*(3), 222–232. doi:10.1002/cae.20043

Rashid, T., & Asghar, H. M. (2016). Technology use, self-directed learning, student engagement and academic performance: Examining the interrelations. *Computers in Human Behavior*, *63*, 604–612. doi:10.1016/j.chb.2016.05.084

Rattadilok, P., Roadknight, C., & Li, L. (2018). Teaching students about machine learning through a gamified approach. *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering*, 1011–1015. 10.1109/TALE.2018.8615279

Schrader, C., & Bastiaens, T. (2012). The influence of virtual presence: Effects on experienced cognitive load and learning outcomes in educational computer games. *Computers in Human Behavior*, *28*(2), 648–658. doi:10.1016/j.chb.2011.11.011

Shams, L., & Seitz, A. R. (2008). Benefits of multisensory learning. *Trends in Cognitive Sciences*, *12*(11), 411–417. doi:10.1016/j.tics.2008.07.006 PMID:18805039

Suits, B., & Hurka, T. (2005). *The grasshopper: Games, life and utopia*. Peterborough: Ontario Broadview Press.

Sutton-Smith, B. (1986). *Toys as culture*. Gardner Press.

Tanenbaum, J., Gardner, D., & Cowling, M. (2017). Teaching pervasive game design in a zombie apocalypse. *Extended Abstracts Publication of the Annual Symposium on Computer-Human Interaction in Play*, 165–177.

Turner, J. B. (2004). *Your musical child: Inspiring kids to play for keeps*. Milwaukee, WI: Hal Leonard.

Wilkinson, P. (2015). A brief history of serious games. In *Entertainment Computing and Serious Games International Gi-Dagstuhl Seminar 15283, Revised Selected Papers*. Dagstuhl Castle, Germany: Springer-Verlag New York Inc.