

Literature Review

Catherine Yim

cyim@oxy.edu

Occidental College

1 Problem Context

In the United States, women make up only 26 percent of Computer Science and Mathematical Science professionals. [17] This means that there are three times as many men than women in the STEM field. More specifically, in computer science related positions, women occupy a mere 18 percent as of 2015. Evidence shows that under-representation leads to a narrow range of perspectives that diversity brings to research. Diversity expands research possibilities, addresses biases, and produces better outcomes. [13] Conversely, the lack of diversity limits research possibilities, leaves room for biases, and produces inferior outcomes. In addition, the lack of workplace diversity is, plainly, inequitable. [17] The computer science field includes some of the fastest-growing and highest-paying jobs. This means that women are under-represented in one of the biggest, flourishing, and lucrative industries of today. [13]

Women's participation in computer science drastically decrease from high school to college and then again from education to the work force. [3] This phenomenon is referred to as the Pipeline Shrinkage. There could be many reasons for this shrinkage. One contributor is social factors. Girls are discouraged from pursuing computer science, leading to a lack of self-confidence in the subject matter. This does not group well with the fact that programming is not a very welcoming topic to begin with. Coding can be intimidating and hard. This coupled with the discouragement that girls face in regards to computer science possibly plays a role in the creation/preservation of the Pipeline Shrinkage.

Along with this shrinkage trend, throughout time, the population of women computer scientists dropped from 37 percent to 22 percent, from 1995 to 2022. [2] The already problematic gap continues to widen. As the industry grows and the gender gap continues to widen, it is becoming increasingly more important to address the diversity issue and ensure that all people have equal opportunity to join the computer science field if they wish to do so.

2 Technical Background

2.1 Why an Edutainment Game?

As the name suggests, an edutainment game is an educational game usually created for one of two purposes. The first purpose is to help the user improve their skills or knowledge of a particular subject, and the second is to peak interest or introducing the user to a new and completely foreign concept that may seem a little intimidating at first. My project is going to focus on trying to make the user more comfortable with the idea of programming and try to make it more approachable to a wider audience (i.e. women and other under-represented demographics).

Digital edutainment games are largely successful worldwide because they utilize the motivational aspects of a game. It is a widely accepted idea that the act of "playing" is a positive human experience and that edutainment games use some of the characteristics of play can be formalized into a structured activity promoting learning. [6] They do so by intriguing curiosity, challenge, and fantasy [10]; influencing the gradual unfolding of the game through provision of feedback loops and mastery of subject [14]; allowing the user to interact with the game on an (almost) personal level by creating an identity through an avatar. [4] These aspects provide a flawless/seamless gaming experience that promotes not only creativity but an enjoyable learning experience.

More specifically, there are many ways in which an educational game uses proponents of "play" to make learning more effective, and these attributes are clearly outlined in Gee's article, "Deep learning properties of good digital games: How far can they go?" Educational games should have goals, competition, rules, choices, challenges, fantasy, fidelity, and context. Goals of the game should correspond with the learning objective. There should be an element of competition, either against oneself or another) as an incentive to do well. The rules of the edutainment game should mirror real-life constraints as to create a sense of difficulty and challenge. There should be decision-making proponents of the game that let's the player make choices to expressive choices "such as building an avatar that influence motivation but not learning, strategic choices such as level

of difficulty that impact the outcome of the game, and tactical choices about how to play the game in a 'branching' game with more than one path." [9] The game should also intrigue curiosity, challenge, and fantasy [10] which influenced the gradual unfolding of the game through provision of feedback loops and mastery of subject [14]. This should allow the user to interact with the game on an (almost) personal level by creating an identity through an avatar. [4] These aspects provide a flawless/seamless gaming experience that promotes not only creativity but an enjoyable learning experience. In short, games are fun and edutainment games have adopted its features to also make learning enjoyable and easy. This makes edutainment games a powerful and widely used tool that is (as we established earlier in the paper), not accessible to certain groups. Groups that would, arguably, benefit the most from access to educational games.

2.2 Game Choice

The goal is to make the game approachable/enjoyable and educational. The reason behind my choosing *Flappy Bird* [1] as a framework for the gaming component of my project was due to its popularity. It's popularity stemmed from its simplicity. It was easy for users to engage with the game since it did not require its players to master a complex set of controls, understand weapons, or memorize anything. This resulted in early wins and instant gratification. Additionally, its simplicity will also grant the players of my game the leisure to focus on much more than just the game (learning how to code). I'm not sure if my project will result in a copyright infringement. If this is the case, then I will create a new game design that is simple and offers instant gratification. Hopefully it will produce the same pleasure effect.

2.3 Language Choice

Javascript was chosen as the programming language of choice because it is a widely used coding language with many applications. The vast majority of browsers are based on Javascript and is a hugely in-demand skill. It is also a relatively easy language to learn and relatively accessible. It does not require the user to download anything to use it. If the user were to further their Javascript learning or reproduce my project, all they need is a web browser. For example, if the user has Chrome, they can simply open up Chrome, navigate to Developer Tools, and they can start coding right away.

2.4 Why a Customizable Game?

A customizable game is more accessible. Players with disabilities can benefit greatly from customization. For ex-

ample, this game prompts the user to change the size and color of the avatar. This can be useful for players with visual impairments, such as limited vision or colorblindness. Another example is that the game allows the player to change the speed of the game. This can be helpful to players with cognitive disabilities that face difficulties with time-sensitive matters.

The customization of a programming game in particular allows the user to gain practice while creating. It lets the user make the game more personal. Making the game customizable is essentially giving the user the freedom of creativity and increased interactivity. This process could or customization, especially avatar customization, in a game is seen as attractive because it establishes a connection between the user and the game. It is meant to increase player immersion in the computer game narrative, and the user is meant to feel empathy towards the avatar and identify with it. [14] This allows the user to immerse themselves in the game, making the game more enjoyable.

3 Prior Work

There's a vast collection of computer science-related edutainment games that can be accessed through the internet. Most games help the user improve their skills or knowledge of a particular subject or language, whether it be the basics of programming (e.g. *Move Code Lines* or *Comet 64*) or something more fun and challenging (e.g. *Screeps* or *Robocode*). Some games even help people train their brains to practice logic-based thinking in order to properly communicate with computers. Through surfing the web, I've found a few games/projects that align with my goal of increasing the approachability of programming to a wider audience. An example of a simple coding edutainment game is *Kodable* [12]. *Kodable* is a coding curriculum geared towards elementary students. It allows users to familiarize themselves with JavaScript by customizing their own "Fuzz" and taking them on digital coding game adventures.

There is also *Scratch Imagine, program, share*, which is a high-level block-based visual programming language and website. It is an educational tool geared towards children and is available in more than 70 languages. Due to its effectiveness and accessibility, *Scratch* is widely used by standard and nonstandard education systems.

Another similar project that I found was conducted by Girls Who Code. Girls Who Code is a non-profit organization dedicated to promoting gender diversity and aiding young women who are historically underrepresented in computer science fields. The organization teamed up with rapper, Doja Cat, to make her music video for "Woman" user-interactive. They created an interface that lets viewers change images from the video using HTML, CSS, and JavaScript.

4 Methods

4.1 Project Idea

I will start by creating a single web application game/code editor using JavaScript. Within the code editor, there will be a pre-coded game that's mechanics resemble those of *Flappy Bird*. [1] The user starts with a dog as their avatar and simple green lines for vines. The premise of the game is that there is Sue, the Dog, who is missing her pups. She has to find and collect them before nightfall by traversing through a jungle. The first stage of the game would be to let the user customize Sue's color and dimensions. The rest of the game will come in stages.

The user will play until Sue comes across her first pup. As she meets and combines with her pup, their dimension becomes larger. The guided to change the size of the Sue and Pup1 since they are currently too large to avoid the vines (which Sue must do in order to go through the jungle). This is the user's first challenge. They must re-adjust the size of the Sue to make the game playable. Just in case the player gets stuck, there will be a hint option in the corner that highlights the part of the code that they must edit. Once successful, the user is allowed to continue playing the game. Until they come across the second pup. The user must resize the pack once more so that the group of three dogs can traverse through the jungle, and then the second challenge is presented. The vines grow to be too long, spanning from the top of the screen all the way to the bottom, making the game impossible once more. The user is now challenge with the task of shortening the height of the vines, given a hint. Once the third and last pup is acquired, the user must once more readjust the size of the pack and shorten the growing vines once more. The final challenge will be that the it becomes nighttime. This means that the background became dark. The user is now tasked with the challenge of changing the background back to white so that Sue and her pack is no longer "scared" and can make their way out of the jungle safely. Now that the user has been introduced to the concept of altering the code to change the avatar dimension, vine dimension, and background color with guidance, their final and most difficult is presented to them. Sue forgot that she, in fact, has four pups and missed one along the way. Sue must go back into the jungle and find her remaining pup with the same obstacles as before. Once the game is over, the player is given the option of free-play. They will be able to prompted to change the attribute of their choosing as they see fit. I will also create a GitHub repository with images (.png files) that would aid users to customize their game to resemble *Flappy Bird* [1] graphically.

4.2 Components of Game

4.3 Set-Up

I will start by creating Visual Studio Code a create the name-space for the game. To avoid polluting the global name-space. I will be using the JavaScript canvas API to do this. Then, I will add some basic object that will allowed me to create basic shapes like circles and rectangle and preform some basic functions like screen-clearing and text adding This is meant to make it easier for myself (so that I don't have to remember the exact canvas API calls), but also for the player when they later get to customize their avatar in the game. We create the story-line and flow of game in the set-up as a framework of the game.

4.4 User Action to Input

There are a lot to consider when making a mobile game, however, a large part of it being getting the input to effect output. For example, in this web app, the user to be able to "jump", they must be able to tap the screen and have that action interact with the actual game. We will call this a touch event. We create an array for touch events, each element of which contains touch coordinates and other data. I'm not entire sure on the specifics of that yet. The single web application will display a game/code editor using JavaScript. Within the code editor, there will be a pre-coded game that's mechanics resemble those of *Flappy Bird*. This is where the user can edit the code to alter the game and complete the challenges. The changes that the user makes to the code is another event that the has to be passed onto the input object. The loop is where this happens. We then create a game loop that polls the user input, updates the web-app's characters and collisions, and repeats.

5 Evaluation

5.1 Participation Selection

Ideally, I would like to recruit participants who do not have any experience with coding. In order to do so with as little bias as possible, I will get all send out an email to the Occidental College email chain, asking people to play the game. A possible issue with this form of participant recruitment is that there is a non-response bias. Students/faculty who are open to the idea of playing a programming game would most likely be the ones to respond to this email. Another potential issue is that no one might be willing to participate. This is extremely viable since the process will be extensive and not convenient.

5.2 Evaluation Metrics

As stated before, the goal of this project is to make computer science less intimidating. For this reason, a meaningful metric of success would be an indicator of increased comfortability with the subject of computer science before and after playing the game. I will be creating a survey that asks participants the following questions before the playing the game: 1) From a scale of 1-10, how intimidating is the concept of programming? 2) Yes or no: Do you have any experience with programming? 3) If so, from a scale of 1-10, how confident are you in your programming abilities? 4) If not, from a scale of 1-10, how confident are you in your programming abilities? and after playing the game: 5) From a scale of 1-10, how enjoyable did you find the game? 6) Yes or No: Do you feel as though you've learned something? 7) From a scale of 1-10, how confident are you in your programming abilities?

6 Ethical Considerations

Disabilities and other divergences are often not accounted for in the designing process of many games. Users with low vision or blindness, motor/physical disabilities, and/or mental/developmental disabilities may find themselves at either a disadvantage or unable to play the game altogether.

6.0.1 Motor/Physical Disabilities

Just as barriers that cognitively disabled players encounter can be addressed with design alterations, ones that physically disabled users face can be tackled with software/hardware changes or accommodations. Some common features that game developers implement to make the game more accessible to physically disabled players are customizable controls, a basic interface mode, and support for special devices. [7]

Customizable controls are already a common feature that many games have. It allows the user to remap controls to ones that they are able to use and can be very beneficial to people with limited mobility. For example, arrow keys on a computer can easily be substituted for WASD, which is the left-most control scheme on a keyboard. This is not only useful for players with limited mobility on their right hand, but also for users who are left-handed.

Typical gaming devices support the standard mouse, joystick, or gamepad. These standard gadgets could prove to be difficult for people with mobility issues to use. Improving hardware to support special devices would allow a larger pool of players to enjoy a larger number of games.

Some games have complex interfaces. Many physically disabled players could find it difficult to navigate through

such a game even with customizable controls or special devices, therefore, many multiplex games have an optional simplified interface mode with just the basic controls. Although the full features are still available, this simplified mode shows only the most used and necessary controls. This simplified interface mode is meant to minimize physical movement required in game-play.

6.0.2 Blind/Low Vision

Blind and low vision users typically have trouble with visual signals. In other words, they have trouble perceiving feedback provided by the game. [7] Two alternatives to visual stimuli are audio stimuli and haptic feedback.

Along with receiving output, blind users often struggle with providing input. Similar to physical disabilities, visual disabilities often make it difficult to use standard controls/controllers. Blind or low vision can be accommodated with the implementation of customizable controls or improved hardware to support special devices. For example, computer games that rely on mouse clickers require the user to know exact locations to click (which can be difficult with the absence of visual abilities). Games that are controlled with a mouse can have a customization option of switching to a keyboard operated game so that the user is not required to input commands based on a specific screen location.

6.0.3 Hearing Disabilities

Players with auditory impairments experience difficulties when effects, essential information, parts of the plot, or other essential information is conveyed auditorily. [7] These types of barriers could be resolved through the use of subtitles.

Subtitling and close captioning are one of the most popular accessibility options in games. Although the vast majority of games offer subtitles, they are usually insufficient. Current game subtitling practices do not follow widely applied audiovisual translation (AVT) industry standards. [11] Some common mistakes that game developers make when creating subtitles are small texts, unclear fonts, and lines that are difficult to digest.

6.1 Cognitive Disabilities

Computer software have been gaining attention as a tool to potentially help children and adults with developmental disabilities to "learn, communicate, play, and be more independent in their lives." [5] Naturally, a large part of the discussion surrounding the design of these games is regarding accessibility of the computer applications for individuals with cognitive impairments.

There are a variety of different cognitive disabilities, and the barriers that these users may face vary depending on

their specific condition and individual abilities. [15] Typically, game users (players) with cognitive disabilities have problems with the design, content, and mechanics of the game. [15] This includes aspects like incompatible language registry or insufficient time allocation for decision making given a stimulus. Game developers employ to reduce or eliminate this accessibility issue include avoiding possibly triggering features by steering clear of strongly contrasting colors, avoiding pop-ups, reducing time constraints, the amount of stimuli or input, and providing alternative difficulty levels. [15] For example, video-game *Celeste* [8] has a feature in its assist mode that allows the player to change the speed of the game. Some games even go as far as to allow the user to alter the difficulty of the game, down to the specifics of enemy damage and attack frequency.

There is an additional layer of consideration that goes into creating an educational game in the case of cognitive disabilities. They are especially important to consider when creating an edutainment game because the ways on these disabilities typically effect the way that people learn. For example, children with down syndrome often have difficulties with repetition repeated practice of functional activities because of movement limitation, attention deficit, or cognitive impairments or a lack of intervention context variability. [18] Many gaming frameworks, especially Virtual reality (VR) games, requires the user to interact with the scenario or environment to a strong intensity and positive sensory feedback. [18] This can be very triggering for user with down syndrome and other developmental disorders since many cognitive disorders cause stimuli sensitivity.

6.2 How this Pertains to my Project

The design, content, and mechanics of my project are fairly simple. It should not prove to be overwhelming for cognitively disabled users as it does not display quick flashes, harsh color contrasts, or regular moving patterns. Unlike games that have vibrant animations and brisk movements, my proposed game starts with simple colors and movements the game is itself is very customizable. Similarly, my game is already very limited in terms of controls and does not have a complex interface, and is therefore somewhat accommodating for people with impaired motor skills. Unlike games with more complex interfaces that present clear difficulties for those with impaired motor skills even with accommodating alternative control schemes, my game is usable as long as the user can perform basic motor functions such as clicking a mouse and typing a few lines of code. In addition, unlike games which rely heavily on auditory cues (such as first-person shooters), my proposed game is largely visual and lacks essential auditory

cues, mitigating the disadvantage that those with hearing disabilities suffer in using it.

However, my project is far less accessible to those with other physical impairments. Groups with low visual ability in particular would have difficulty using my app, as the gameplay and usage is heavily dependent on visual cues. While the information for the coding portion of the game could be communicated via text-to-speech, it's unclear how the game could relay spatial information for the game-playing portion through sound or other means. The solution of utilizing substituting haptic feedback and auditory cues for visual cues would be largely unavailable as a remedy for my app, as would hardware that lets the user experience the map of the game with a physical, moving model. These fixes would be much too costly and not a viable solution for my senior project.

7 Timeline

May 15, 2022

Goal: Have a solid story-line and game-flow idea.

June 1, 2022

Goal: Start learning about UX/UI.

June 15, 2022

Goal: Set-up the framework for app.

July 1, 2022

Goal: Continue to set-up the framework for app.

July 15, 2022

Goal: Create Github with resources.

August 1, 2022

Goal: Write functions that feed into input.

August 15, 2022:

Semester starts 02/25

Goal: Continue to write functions that feed into input.

September 1, 2022

Goal: Write game portion of the project.

September 15, 2022

Goal: Make sure game is working.

October 1, 2022

Goal: Create evaluation survey

October 15, 2022

Goal: Get some evaluation results.

November 1, 2022

Goal: Tie up loose ends and work on final paper.

November 15, 2022: Poster due

December 1, 2022: Poster presentation 12/05

December 15, 2022: Final paper and code due

References

- [1] .Gears. *Flappy Bird*. [Arcade]. 2014.

- [2] *About Us*. Dec. 2021. URL: <https://girlswhocode.com/about-us>.
- [3] Jessica D Bayliss and Sean Strout. “Games as a” flavor” of CS1”. In: *Proceedings of the 37th SIGCSE technical symposium on Computer science education*. 2006, pp. 500–504.
- [4] Jim Blascovich and Jeremy Bailenson. *Infinite reality: Avatars, eternal life, new worlds, and the dawn of the virtual revolution*. William Morrow & Co, 2011.
- [5] Amal Dandashi et al. “Enhancing the cognitive and learning skills of children with intellectual disability through physical activity and edutainment games”. In: *International Journal of Distributed Sensor Networks* 11.6 (2015), p. 165165.
- [6] John Dewey. “Science as subject-matter and as method”. In: *Science* 31.787 (1910), pp. 121–127.
- [7] Markku Eskelinen. “The gaming situation”. In: *Game studies* 1.1 (2001), p. 68.
- [8] Extremely OK Games. *Celeste*. [Arcade]. 8.
- [9] James Paul Gee. “Deep learning properties of good digital games: How far can they go?” In: *Serious games*. Routledge, 2009, pp. 89–104.
- [10] Thomas Malone. “What makes computer games fun?” In: *Proceedings of the Joint Conference on Easier and More Productive Use of Computer Systems.(Part-II): Human Interface and the User Interface-Volume 1981*. 1981, p. 143.
- [11] Carmen Mangiron. “Subtitling in game localisation: a descriptive study”. In: *Perspectives* 21.1 (2013), pp. 42–56.
- [12] Jon Mattingly. *Programming for Kids*. URL: <https://www.kodable.com/>.
- [13] Elinor Ostrom. “The Difference: How the Power of Diversity Creates Better Groups, Firms, Schools, and Societies. By Scott E. Page. Princeton: Princeton University Press, 2007. 448p. 19.95 paper.” In: *Perspectives on Politics* 6.4 (2008), pp. 828–829.
- [14] Hua Qin, Pei-Luen Patrick Rau, and Gavriel Salvendy. “Measuring player immersion in the computer game narrative”. In: *Intl. Journal of Human-Computer Interaction* 25.2 (2009), pp. 107–133.
- [15] Javier Torrente Vigil et al. “Towards a low cost adaptation of educational games for people with disabilities”. In: *Computer Science and Information Systems* (2014).
- [16] Unknown Unknown. *Imagine, program, share*. URL: <https://scratch.mit.edu/>.
- [17] Jennifer Wang et al. “Gender differences in factors influencing pursuit of computer science and related fields”. In: *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*. 2015, pp. 117–122.
- [18] Yee-Pay Wuang et al. “Effectiveness of virtual reality using Wii gaming technology in children with Down syndrome”. In: *Research in developmental disabilities* 32.1 (2011), pp. 312–321.