

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. [7] 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. [5] 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. [7] 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. [5]

Women's participation in computer science drastically decrease from high school to college and then again from education to the work force. [2] 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. [1] 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

For my senior project, I'm proposing to make a web-app. A web-app (or web application) is software that runs on a

web browser. Web apps are useful because they can be presented to the user's device as long as it is connected to an active network connection. My web-app will embody an edutainment game that allows the user to customize their own 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. They do so by intriguing curiosity, challenge, and fantasy [4]; influencing the gradual unfolding of the game through provision of feedback loops and mastery of subject [6]; allowing the user to interact with the game on an (almost) personal level by creating an identity through an avatar. [3] 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.

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. 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. 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 Game Choice

The goal is to make the game approachable/enjoyable and educational. The reason behind my choosing Flappy Bird 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.

4.2 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.

4.3 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 the avatar can also make the game personal. An avatar in a game is seen as attractive because it establishes a connection between the user and the game. Players are meant to feel empathy towards the avatar and identify with it. This allows the user to immerse themselves in the game, making the game more enjoyable.

4.4 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. The user starts with a black ball as their avatar and black blocks instead of pipes. Every time the user loses the game, they’ll be asked the question: “What part of the game would you like to change?” They will be given the options of Avatar, Blocks/Obstacles, Background, or None to choose from, depending on the button they choose, the corresponding portion of the code will be highlighted (and along with it, more information on how to change it). The user/player will then be 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 graphically.

5 Evaluation

6 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.

6.1 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?

6.2 Evaluation Results

7 Ethical Considerations

References

- [1] *About Us*. Dec. 2021. URL: [https : / / girlswhocode.com/about-us](https://girlswhocode.com/about-us).
- [2] 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.
- [3] Jim Blascovich and Jeremy Bailenson. *Infinite reality: Avatars, eternal life, new worlds, and the dawn of the virtual revolution*. William Morrow & Co, 2011.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.