Games For Teaching Children: WPI Part-Time Hero

Alexander Bell, Haley Hauptfeld, Javier Marcos, Linda Puzey, and Oliver Rayner

Worcester Polytechnic Institute

Advisors: Therese Smith and Jennifer deWinter

March 18, 2020

A Major Qualifying Project Submitted to the Faculty of the WORCESTER POLYTECHNIC INSTITUTE in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science by May 2021

Abstract

The goal of this project is to make a video game that can teach children (ages seven to nine) educational topics in the fields of mathematics and computing. In our game, *WPI Part-Time Hero*, the player controls a superhero who takes on tasks and challenges in the form of five minigames. Each minigame is focused on a specific topic in computer science or mathematics. In order to achieve our goal, we conducted testing that focused on the effectiveness of our game, in regards to how the game engages its audience. Specifically, we focused on studying the benefits of active learning, as well as reducing gender biases when educating children in the fields of computer science and mathematics.

Acknowledgements

We would like to thank both of our advisors, Therese Smith and Jennifer deWinter, for the time and effort that they gave us throughout the course of this project. Specifically, we would like to thank Therese Smith for providing guidance in the structure of this project. We would like to thank Jennifer deWinter for her informative advice and knowledge in the field of game design. In addition to our advisors, we would like to thank Ruth McKeough for taking the time to meet with us in order to gain insight on how to properly conduct research through the Institutional Review Board. We would also like to thank our artist, Christian O. Adler, for his talented efforts in creating the art assets to make our game entertaining and engaging. Lastly, we would like to thank all of our participants that played our game for testing purposes. We appreciate the time that everyone dedicated towards assisting this project.

Table of Contents

Chapter 1: Problem Description

As society becomes more dependent on technology, it becomes even more important to get students interested in the computer science field to meet this growing demand. However, many schools are unable to teach their students computer science, due to a limited number of qualified teachers. In 2018, a study by *Horizon Research, Inc.* showed that only 26% of elementary schools in the United States offer education in computer science (NSSME, 2018). Therefore, this discourages students from pursuing the subject later in life because of lack of exposure. Similar to learning a language, the more exposure a student has at a young age to that language, the more likely they are to learn it. Sandra Crosser (2008), a professor at Ohio Northern University, found in her study on language learning that, “the brain is most plastic, or flexible, in young children” (para 13). This shows how introducing computer science at a young age is vital to children absorbing the subject; making it vital to teach them at this time.

Fortunately, some of the topics taught in elementary and middle school are also foundational computer science skills. Computer science is heavily related to mathematics and the majority of schools in the United States start teaching math early. An important topic from math that is vital to computer science is using loops. A loop is written to complete the same task multiple times with different inputs. Loops are used within functions in math. Another subject from mathematics in computer science is sorting. Lists or arrays are primarily used when sorting data in computer science. The ability to distinguish between two different items is taught as early as preschool. The above-mentioned are only a few topics in both subjects, but there are many more that will be covered later in the paper.

Despite this, many students reject pursuing computer science. The Kapor Center conducted a poll which showed that in “California, the most populous U.S. state with 1.9 million high school students, had only 3% of its students enroll in CS in 2017" (Scott, 2019, p. 4). One reason for this is that they may not know that both math and computer science are related. Thus, they gain the preconceived notion that they have no background in the subject. Another reason is that students can associate a bad learning experience with the topic, discouraging them from pursuing the subject any further. A study by Friedman showed that "a bad teacher can damage a student's confidence and self-perception" (2018, para. 8). This causes students to doubt their abilities and to dismiss the subject. Similarly, studies have that if there are no qualified computer science teachers within a school or program, then many students can become intimidated by the subject (NSSME, 2018). In addition, computer science is an intimidating subject to learn because there are over 700 different programming languages in existence. Ph.D. Susan Weinschenk says in her book that "people who have too many choices will not choose at all" (2011, p. 93). The many options of computer science may overwhelm newcomers.

Another consideration regarding challenges in computer science education is gender biases. It's important to consider how our game is designed to be inclusive to both boys and girls. This is important because females are currently underrepresented in STEM (Science, Technology, Engineering, and Mathematics). Bieg's (2015) found the following:

> Meta-analyses of international research typically show small or no gender differences in math performance. Research on math-related attitudes, however, has found individuals' attitudes regarding mathematics to vary to a much higher degree, with girls tending to report less positive attitudes and higher levels of anxiety regarding mathematics than boys" (p. 2).

These reactions have been found to deter girls from pursuing careers in topics such as math and computer science (Huang, 2019). We planned for our game design to lower mathematics/computing anxiety for girls by intentionally making any character in the game gender neutral.

Besides preparing students for the future, there are many other benefits to teaching computer science at a young age. One such reason is that it teaches children problem-solving skills. In the future, they will have to be able to understand the problems they encounter and find a solution. Problem-solving is a vital skill in many aspects of life. Computer science uses code to accomplish a task or solve a problem. The task could be as simple as having a program print 'Hello World', or it could be as complicated as controlling a robot. Whatever the need may be, the programmer is the one who devises a solution.

The challenges of learning computer science are made more appealing to a younger audience if they come in the form of a video game. Educational games are not a new concept. Many businesses such as *JumpStart* and *The Learning Company* continue to make educational games for children. These games are great for teaching typical academic topics such as math and English. However, from the years 1971 to 2013 there have been no educational gaming softwares marketed to teach computer science ("List of educational video games", 2020). Many companies opt to teach it using online games, such as *Hour of Code* and *Codecademy*. The main issue with these games is that they inhibit growth by providing too much support to the students. Anthony Grant notes in his article that "Codecademy courses do not teach you to think like a coder. Rather, it teaches you the basics of a number of programming languages without much instruction on how you'd apply them to real-life problems" (2019, para 7). It does not teach the computer science problem-solving skills, because it provides step-by-step instructions for students to follow. Additionally, they may neglect the foundational skills that are required before one can understand coding, which include basic math skills, data types, functions, Boolean logic, and sorting.

One of the hardest aspects of learning to program is understanding how the code translates to action. Fortunately, tools currently exist that are useful for helping children understand this. For example, the *Root Coding Robot* and *LEGO Mindstorm* help students learn basic coding, and see how the code that they wrote becomes real world actions (LEGO, 2021) (iRobot, 2021). Unlike the previously mentioned educational games, *LEGO Mindstorm* and the *Root Coding Robot* are both robots that run off code from their built-in applications. One of the reasons that both robots are so effective is because they are hands-on tools. When students send code that tells the robot to turn left, they can physically see the robot doing that. These tools also offer more freedom to the user. Typically in educational coding softwares, there is only one solution that the software will offer, whereas the *Root Coding Robot* and *LEGO Mindstorm* offer numerous options to accomplish a goal. While these devices are highly effective, the drawback is that they are very expensive. The *Root Coding Robot* and *LEGO Mindstorm* both cost over two hundred dollars, leaving many schools or parents unable to afford them.

The goal of our project is to create an engaging, affordable, educational tool for teaching second and third graders computer science. It is easy for students to be intimidated by a large file filled with code, and at the second and third grade level, they are not proficient at typing on a computer's keyboard yet. To combat this, we opted to teach students only the fundamental skills they need to start coding without using a keyboard or complicated files. These include basic math, data types, Boolean logic, and loops. Basics mathematics includes addition and subtraction. Data types are a way of classifying different kinds of values in computer science. Common data types include integers, strings, and Booleans. Integers refer to whole numbers;

strings are words or sentences; Booleans are true or false values. Loops are used to repeat the same line of code multiple times.

Learning these topics is not always engaging or entertaining for students, so we decided to make a game out of them. After looking at the winners of the Kids' Choice Awards, we came to the conclusion that our game should be themed around superheroes due to the popularity of the genre (Kids' Choice Awards, 2017). We chose to make a series of different minigames, each one covering a specific skill. Our game has five minigames, one made by each group member. The game *Future Teller*, helps teach Boolean operations and logic. Similarly, the game, *Superpower Selector*, is meant to educate the player about Boolean values. Next, *Stop that Fire*, covers the topic of different data types and distinguishing between them. The minigame game *QuickGig*, was created with the goal of teaching the concept of loops. Lastly, *Lightning Change*, teaches how to do basic addition and subtraction. We describe the methodology of each game in Chapter 3.

Chapter 2: Related Work

2.1: Biases in Gender When Learning Computer Science and Math

Studies have found that many students face anxiety when learning topics such as mathematics and computer science (Huang, 2019). Since the 1980s, both subjects have become a male-dominated field, research demonstrates that female students typically experience a greater lack of confidence and increased anxiety when learning these subjects. (Bieg, 2015) The characters in *WPI Part-Time Hero* are gender neutral in order to encourage female students to gain more confidence and reduce anxiety when learning these topics.

Stereotypes in the fields of math and computer science have a great impact on a student's perception and confidence in their abilities to learn the subject. A study of children from Germany "show an alarming effect of stereotype endorsement for girls who, due to inaccurate beliefs in gender differences in math ability, are at risk of believing they are more anxious than they report feeling in mathematics domains" (Bieg, 2015). A study of children in the Netherlands found that "higher math anxiety was related to lower math performance for girls but not for boys" (Erturan, 2015). It is important to consider the stereotypes and biases behind learning math and computer science so that all students can feel included in the learning process.

Encouraging female students to learn math and computer science was an important consideration when we designed *WPI Part-Time Hero*. Using recommendations from the Institution of Education Sciences (IES), the design of the game purposefully includes a lack of characters to avoid having to decide between male and female characters. For example, one recommendation is to "create a classroom environment that sparks initial curiosity and fosters long-term interest in math and science" (Halpern, 2007). This is accomplished by creating a design that is gender neutral in order to draw the attention of both girls and boys. A study in the U.S. found that "in order to develop girls' interest in a mathematics-related career, it is essential to develop effective strategies to reduce their math anxiety and remove gender stereotypes related to STEM fields" (Huang, 2019). By creating a gender neutral learning experience, female students are more likely to feel more included in the learning experience because they do not see boys or girls represented in the design of the educational game. Rather, they will be instructed from prompts during the game, giving them the idea that gender should not impact how students learn math and computer science.

2.2: How Mindset Affects Learning

Mindset can affect much of what we do in life, and most relevant to our study is how it affects learning growth. In much of her works, psychologist Carol Dweck asserts that there is a continuum between two different mindsets that all individuals lie on (Dweck, 2008, p. 6). At one end of this spectrum lies the fixed mindset (Dweck, 2008, p. 6). A fixed mindset is the belief that an individual has a certain set of traits associated with them (Dweck, 2008, p. 6-7). For example, an individual with a fixed mindset might believe that he/she has a predetermined skillset, like being good at basketball but bad at math. At the other end of the spectrum is the growth mindset (Dweck, 2008, p. 7). This mindset is the belief that individuals can be molded and changed by their experiences in life (Dweck, 2008, p. 7).

Like the names may suggest, a growth mindset allows individuals to rebound from dealing with failure or rejection faster than a fixed mindset (Dweck, 2008, p. 9). When dealing with unfortunate situations, such as a bad midterm or corporate review, an individual with a fixed mindset might claim “I feel like a reject”, or “I’m a total failure”. Dweck states that individuals with a fixed mindset take the event as a measure of their own competence (Dweck, 2008, p. 8). However, people with the growth mindset experiencing the same event might claim “I need to try harder in class”, showing a divide in the thoughts of people with a fixed mindset versus a growth mindset (Dweck, 2008, p. 9). The difference in the two mindsets can be striking when compared to each other.. Often, individuals with a fixed mindset turn outwards or give up as coping mechanisms (Dweck, 2008, p. 8). Individuals with a growth mindset would often cope by looking towards the future, such as by resolving to study harder or ask the teacher for assistance (Dweck, 2008, p. 9). According to this research, it would be better for individuals to have a growth mindset when attempting to learn new concepts. This also includes learning from the game we made.

According to Dweck's research, it is possible to change your own mindset to be a growth mindset. Not only that, but it can be instilled into students at a younger age (Dweck & Yeager., 2019). This research shows that instructors can easily instill a growth mindset into students (Dweck & Yeager., 2019). However, her research showed that many educators do not fully grasp what a growth mindset entails, and thus do not support the growth mindset to its fullest (Dweck & Yeager., 2019). It is important that we reinforce a growth mindset in our players, so that they are more likely to be receptive and learn the concepts we teach them.

2.3: Students Acquire More Skills Through Active Learning

Active learning is a method where students are encouraged to interact in activities in order to learn a subject. An example of active learning is students sitting in groups or in a circle interacting and experimenting among their peers. However, modern schools mostly use traditional, lecture based learning where students are sat at desks facing their teacher and are lectured on the topic. Studies have shown that active learning is a better method in successfully getting students to retain knowledge (Bachelor, 2012. p. 55). *WPI Part-Time Hero* makes children learn actively which will lead them to be more interested in learning school subjects.

In a research article, active learning was applied to classrooms in order to find the benefits of active learning compared to traditional learning. In the study, 338 participants were involved between the two different learning styles across the United States and Canada. There was a 15% decrease in student failures who learned under active learning (Freeman et al., 2014). This is an important statistic for our project, *WPI Part-Time Hero*, because it shows that making learning through interaction helps boost the students' learning capability.

Another study involved different groups of people interacting or not interacting with long and short videos. Eleven minutes or more is defined as a long video. Interaction done with long and short videos was split up into four groups. As seen in Figure 1.

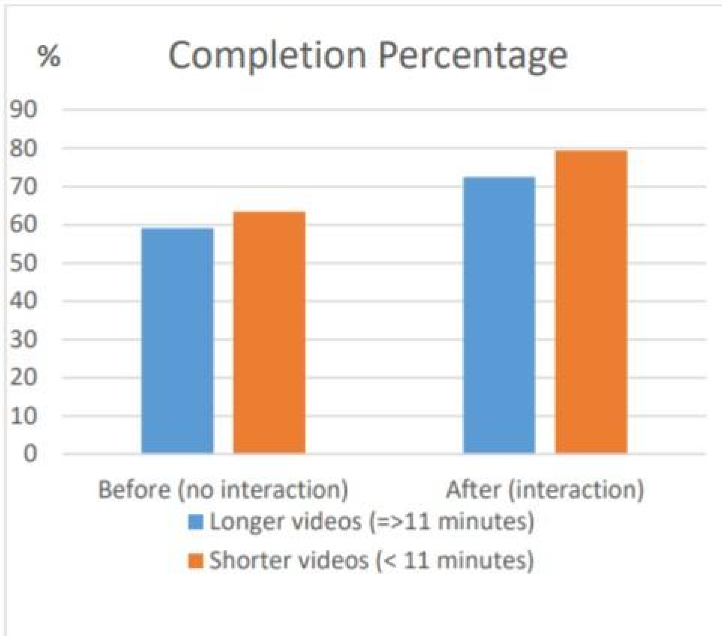


Figure 1: Completion rate of four groups of testers

In the end, the research found that the completion rate was higher for participants who had long and short interactive videos. Interactive long videos had a higher completion rate than short non-interactive videos (Geri et al., 2017). This shows that interactivity is great at getting people to do a task. *WPI Part-Time Hero* focuses on interactivity to increase the completion of our minigames and the students center of focus.

In Australia, a research study took place between two schools to study the effects of educational games in the classrooms. One school was given Nintendo educational games (school A), and the other school was taught traditionally (school B). The math scores went up in school A and stayed the same in school B. Nintendo educational games, such as *Brain Age*, were proven to make the kids in school A learn the material better through interactive games, while school B were not motivated in the same way (Main & O'Rourke, 2011). This study shows that there is an increase in learning when students are having fun interacting with the material. The interactive games within *WPI Part-Time Hero* were designed to combine the right amount of education with gratification. Our educational game has computer science concepts established at its core with fun game design in order to separate the game from traditional learning.

2.4: Technology in the Classroom

Technology has become deeply ingrained in many aspects of our lives, enabling people to work more efficiently. School environments could also benefit from technology, which introduces many opportunities to make teaching easier for staff and learning more engaging and effective for students.

Currently, technology has already been used in classrooms to varying extents, such as watching learning videos in class and slideshows. Several studies, such as the one carried out by Comiskey (2011), show that video notes can be a helpful compliment to teaching. These presentations stand out from usual lectures by providing audio and visual aids, catching students' attention. Comiskey's research also mentions how students themselves asked for "the possibility of them having access to the screencasting software to create their own short videos" (p. 37), which could allow students to reinforce the material even further.

Computer games in a learning environment have also been found to be effective. Short's article (2012) reveals that *Minecraft*, a sandbox building video game, has proved to be an effective way to teach various topics, such as biology, ecology, physics, chemistry and even geometry. This potential has been acknowledged by the game's developer, Mojang, which now provides a platform for teachers and learners which contains an expanding catalogue of lessons as well as other resources. Guardia's research (2019) also suggests that *Kahoot*'s incorporation of gaming elements into the traditional concept of multiple-choice tests has enabled students to better reinforce many different skills, including teamwork, leadership, oral communication and decision-making.

Video games have the potential to make learning more varied and involved, making them a very valuable tool. They can not only help cover various learning topics, but also reinforce important skills for other aspects of life, shaping students into better people. This project aims to develop a game that can help make a game that can leverage its engagement to serve as an effective teaching aid for children in the field of computer science.

Chapter 3: Game Design Methodology

In this chapter, we will be discussing how we created the five minigames and game selection hub within *WPI Part-Time Hero*. We will convey the educational goals of each minigame and how we developed the code to reach that goal. Each member of the team worked independently on one minigame. *WPI Part-Time Hero* was created using the Unity platform and C# scripts. Below, Figure 2 displays specific information regarding who created each game and the objective.

Name of Minigame	Creator	Educational Goal
Future Teller	Alexander Bell	Boolean logic
Superpower Selector	Haley Hauptfeld	Boolean logic
Stop That Fire	Linda Puzey	Data Types
Quick Gig	Oliver Rayner	Loops
Lightning Change	Javier Marcos	Addition/Subtraction

Figure 2: Completion rate of four groups of testers

3.1: Minigame: *Future Teller* Methodology

The game, *Future Teller*, teaches Boolean operations and Boolean logic. For the first version of *Future Teller*, as seen in Figure 3 below, we worked on establishing the narrative of the game and basic gameplay features. The minigame's setting is that the player has the superpower to see possibilities of the future and uses this to change it for the better. However, the future is often very muddled with unnecessary information, so the player must use Boolean logic to determine the final outcome of the event. We chose this setting because it complements our superhero theme. We also set up the first version of the game as a way of making sure that the buttons and other mechanics worked.

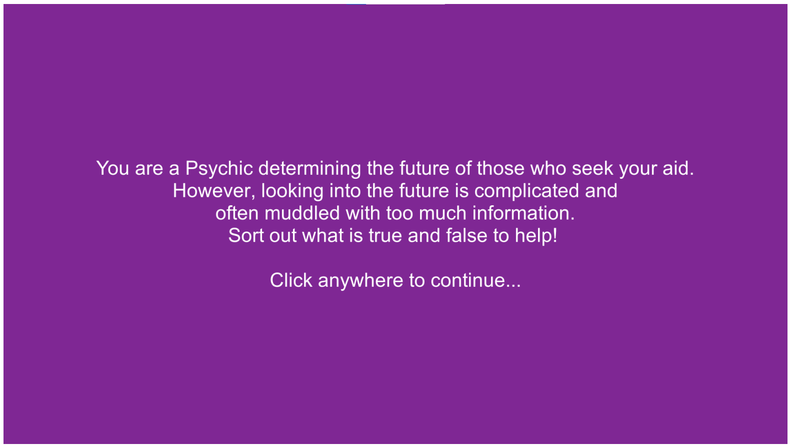


Figure 3: Introduction to an early version of *Future Teller*

The next pass of changes we made to the game was implementing randomly generated levels. We created methods that randomly generate a Boolean expression and internally store the information. Then, we developed the buttons behind the text prompt that help simplify the expression by clicking on a substatement. An example of a substatement would be if the whole prompt was "(True and (True or False))", as seen in Figure 4 below. Then, the substatement within that prompt would be "(True or False)". The user could then click that text to show additional buttons that would replace "(True or False)" with either "True" or "False", depending on the button they click. One issue in this version is that it did not account for dynamic screen sizing. This means that if the user played on differently sized monitors, then text may be cut off, or the margins on the game might be incorrect.. Additionally, the buttons were not quite where they should have been, however they were fairly close. Lastly, sometimes text would be cut off.

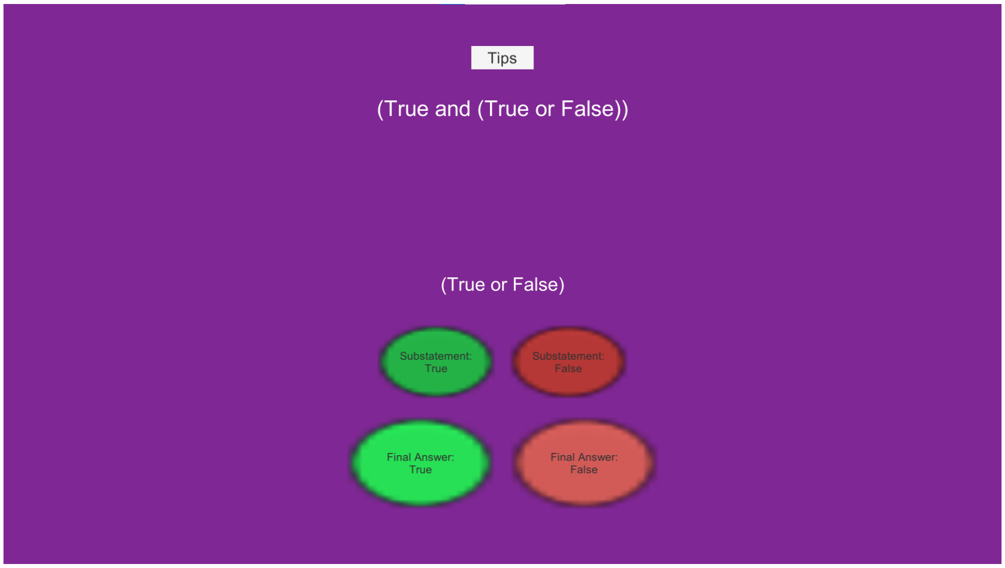


Figure 4: Main gameplay screen

At this point, we tested the games at AlphaFest[1]. Our AlphaFest testing showed that WPI students who had experience with boolean logic were better able to understand and enjoy the game, compared to those who did not. Since our target audience would likely have no knowledge of these concepts before playing our game, this was not a good result. We began to reimagine if there was another way to represent Booleans. One idea that we came up with was to use good and evil cats to represent true and false values respectively. However, there was no clear way of performing Boolean operators on cats. For example, why does a good cat and an evil cat result in an evil cat, and what does it even mean to ‘and’ cats together? We also thought about changing the Boolean operators to make sense with the cat theme. However, then there was not a clear

[1] AlphaFest is an event held once a year and is used to test games that students have been developing. All IMGD students who are making a game for their MQP must participate at AlphaFest, others who have made a game can join if they want to. This was the first time the minigames were set up to be tested, and we did not know what to expect going into the event. More information on AlphaFest testing is discussed in Chapter 4.1.

connection between Booleans and the cats at this point. We decided we needed to take a simpler approach and walk players through step by step as to what each operation means and how to compute their results.

In the next version of *Future Teller,* we added sound effects and background music. We then fixed several existing issues in the gameplay, such as screen resolution complications and button locations. As part of the solution to the AlphaFest problem, we also added a tips page, which explains what the result of each possible operation is. For example, the tips page for the ‘or’ operation has answers for all four combinations of ‘true’ and ‘false’, as seen below in Figure 5. At this point, we tested with one subject in our age range, and it revealed a couple more issues. The first was that the subject did not readily click on the tips page for help. This further reinforces that we need to help players ease into the game by having only one operation and showing them how to answer those questions. The second was that the individual did not attempt to click on the substatement to try and take the question piece by piece.

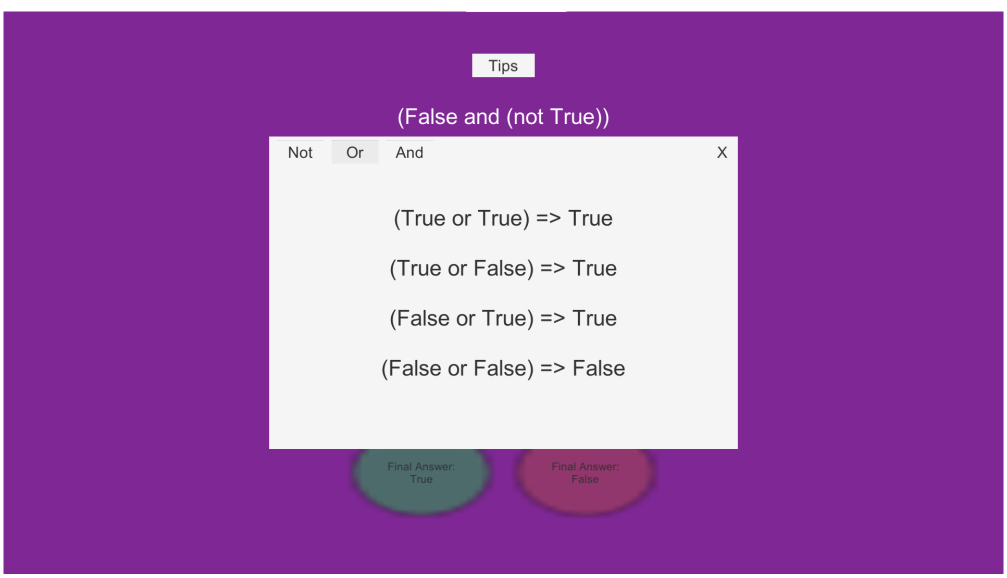


Figure 5: Tips screen showing help for the ‘or’ operation

3.2: Minigame: *Superpower Selector* Methodology

Superpower Selector is a game that is meant to educate the user on the topic of Boolean values in the field of computer science. In the game, users must select the correct superpowers from a grid of choices. The correct superpowers are the ones from the prompt that the superhero character on the screen is displaying to the user. The prompts are written as 'AND' and 'OR' statements that apply to specific superpowers and the color of the background. For example, a user could be prompted to select all instances of 'flight' and 'orange' objects on the screen. The game ends when the user correctly selects all of the superpowers in a 3X3 grid, as seen in Figure 6 below.

Figure 6: Superpower Selector's main gameplay screen

We spent a lot of time researching Unity tutorials and becoming familiar with the UI and the mechanics of the platform. When we started planning how to create the game, we experimented with different types of assets to determine which will lend itself best to the use case of being clicked upon and disappearing. We chose images and developed a grid of nine images that would serve as the background color to each superpower, as seen in Figure 6.

Afterwards, we chose icons from thenounproject.com to serve as representatives of superpowers, which would later change into the assets created by our artist. The superpowers we have included as options are: ‘flight’, ‘invisibility’, ‘strength’, and ‘telepathy’. The possible background colors are orange, green, and purple. We overlaid the superpower icons onto the background colors as images.

Next, we added logic to the game. We wrote a script that develops a prompt that tells the user what to select on the screen in the form of a Boolean statement. There is one script for each cell within the grid on the screen. Everytime the user selects a correct possible cell, it disappears off the screen. Once all of the correct superpowers are selected, the game ends.

Afterwards, we added auditory feedback and music to the game. We retrieved all of the sounds from freesound.org. Whenever the user clicks on a correct superpower, the game plays the ‘correct’ sound. Whenever the user clicks on an incorrect superpower, the game plays the ‘incorrect’ sound.

3.3: Minigame: *Stop That Fire* Methodology

Stop That Fire is meant to teach differentiation between data types such as strings, Booleans, and integers. Integers refer to any whole number, such as 1, 25, or 457. Strings are words or sentences. They are surrounded by quotes, like so: “Dog”, “College”, or “I like Cheese”. Booleans are values that only represent true or false. Data types are used in most programming languages, so being able to distinguish between them is a vital skill.

To keep in line with the superhero theme, the player character is a superhero that can control water. The hero must stop a supervillain who enjoys making buildings burst into flames, and put out the fires before it’s too late. However, the hero's powers are limited in the fact that he can only put out fires in a certain order. If they try to put out the fire in any other order, their water bending skills will be ineffective.

The order corresponds to different data types used in computer science. On the game screen, there are three different fires, all with a different data type example on them. As seen in Figure 7, the data type examples on the fires from left to right are: True, 89, and “Dino”. The top of the screen, also known as the status line, contains a prompt. In the case of Figure 7, it says, “Click on the Integer”. The player must click on the fire that contains the integer to be able to put it out. If they click on another fire, they will be unsuccessful. As stated earlier, an integer is a whole number. In our example, the correct fire to click on would be the one that contains the number 89. After they click on that, the fire will be put out and then the status line will instruct the player to click on a different data type, such as a string or Boolean.

Throughout development, this game was continually improved to be more user friendly. The first iteration of *Stop That Fire* was very simplistic. The main feature we added was the ability to click on the fires, making them disappear. The status line would then instruct the player to the next data type they should click on. One of the main drawbacks for this iteration was the lack of variety in the fires. Each fire only contained one value, and it was the same every time the game was played. In Figure 7, the examples on the fires are True, 89, and “Dino”. In the first iteration of the game, these would be the only examples of the fires. The examples would also remain in the exact same spot as well. In Figure 7, the fire containing “Dino” was always the rightmost fire; the position of it would never change. This iteration had no replayability to it whatsoever. Once the player knew the correct answers, the game lacked educational value because the user could simply memorize the answers.

Figure 7: Stop That Fire's main gameplay screen

For the second iteration, we wanted more variety in the data type examples of the fires. It would be boring if the answers were always the same for the game so we needed to add some randomization. This was done by creating an array, which is a list of data, filled with different numbers, words, and true/false values. The hardest part was making sure that each fire only corresponds to one data type. For example, we didn't want the rightmost and leftmost fire to be numbers. We needed each fire to contain a different data type. This required many different checks within our program to make sure that we were always getting a different data type for the next fire. Each fire had their own code attached to it. Based on the datatype of the first fire, the other fires would change to different data types accordingly. This allowed a great deal of variation in the gameplay. The main issue with this iteration would be that the first fire would almost always be the same data type, usually an integer. The reason that this happened was because all the example data types were stored in one array. There were more integers than any other example in the array. Therefore, when picking a random value out of the array, the player was more likely to pick an integer.

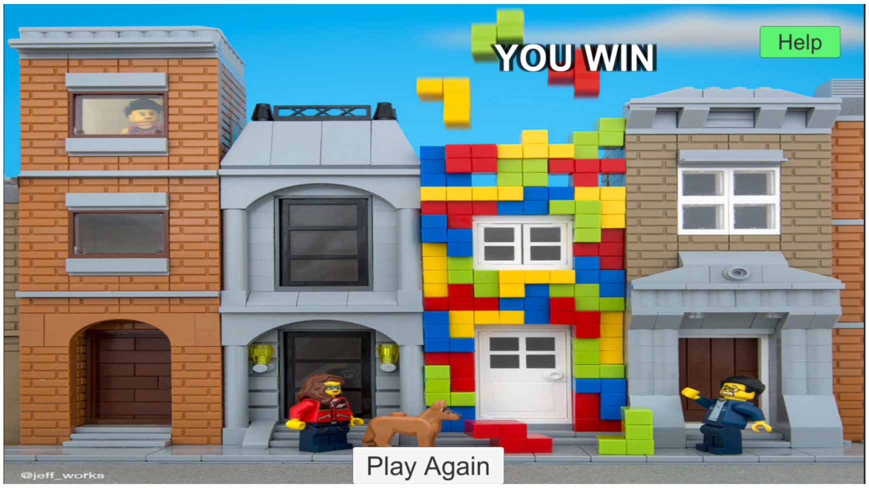


Figure 8: Victory screen, shown after correctly putting out all the fires

In the third iteration, we wanted to make the game even more replayable. Before, the fire examples would always be the same type of data type. Such as the right most fire always being a string data type. The reason that this happened was because all the example data types were stored in one array. Fixing this issue required separating the data type examples into three different arrays, one for each data type: strings, integers, and Booleans. This allowed for much more randomization between all three fires. Now each of them could contain any one of the three data types.

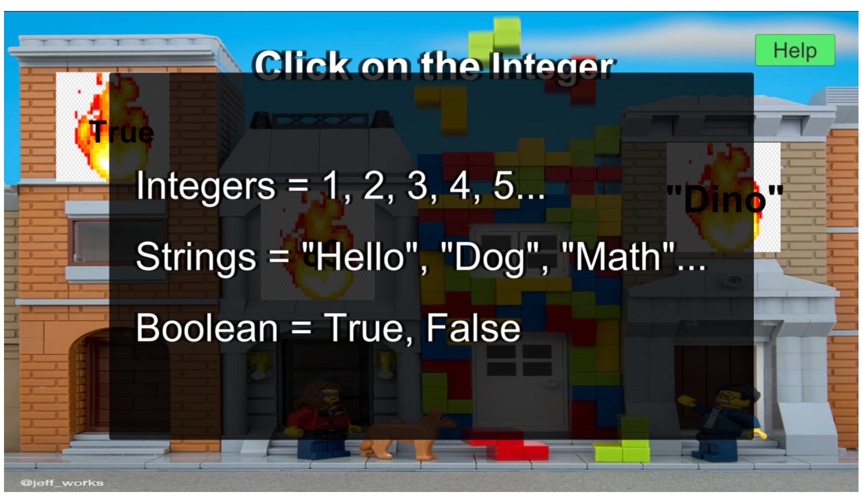


Figure 9: A help menu, showing examples of different data types

After some testing, we noticed that many of the participants had trouble identifying the data types. Therefore, we added a tips screen for the next iteration. The tips screen shows each of the data types and gives examples of what each of them could be. This is a toggleable screen that the player can click on at any point if they need a hint to continue playing, as seen above in Figure 9. We also added a win screen that has a play again button, as shown in Figure 8. Stop that fire is a very short game so we wanted to make sure that the player could easily play again with different examples.

3.4: Minigame: *QuickGig* Methodology

The minigame, *QuickGig*, was created with the goal of teaching how loops work. A green placer is used to organize the food as well as a loop slider that is used to quickly fill how many items in the green area need to be sent out. The background of the game is that a hero needs to have a job in order to make money. Therefore, in the game, the player is tasked with categorizing dishes of food in different quantities. All food assets were made using Adobe Illustrator. One video that was used in our research reviews the game, *Human Resource Machine*, which uses computer science concepts for their gameplay. The person playing the game is a youtuber called *Jacksepticeye*. Throughout the game, he is not using loops (Jacksepticeye, 2015). You can see by the comments that many people were upset with his gameplay. Neonz27 commented, “Was anyone else screaming at the monitor when Jack didn't use loops!?”. These comments were apparent that loops are an essential skill when it comes to teaching students.

Loops are helpful in going through code multiple times without copying and pasting segments of code. Think of loops as multiplying to find the number of boxes in a 100X100 grid and think of not using loops as counting the boxes all the way up to 10,000. It should be clear which method for getting the number of boxes would be faster and more efficient. In this game, the hero calls how many orders need to be made with a slider. Then, food is added to the tray that is on-screen, as seen below in Figure 10. The player is tasked with sending out a certain number of orders within a given time limit.

The game started out with a positive and negative area, which represents liked food and disliked food. Feedback from AlphaFest showed the gameplay was repetitive. Therefore, the game switched to a positive area and a loop slider. The loop slider was implemented in order to have the player understand the concept of loops and increase player interaction. First, the player what food the customer likes and then through the slider to match the order number. When the food requirements and the loop number are met, the scene changes and the player has to beat the clock again. After ten levels, the game will stop and bring the player to a victory screen with a play again button, as seen in Figure 11 below. This idea of having a level limit came from AlphaFest feedback, when players who played the cashier game played for a long time, but did not reach any sort of resolution or victory screen.

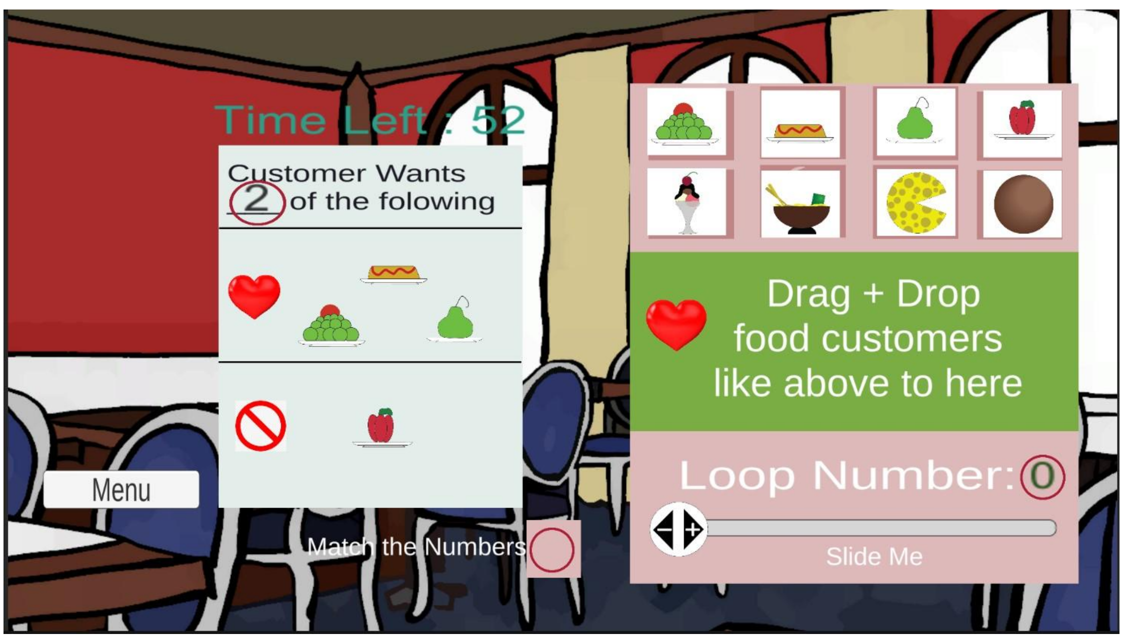


Figure 10: *Quick Gig's* user interface showing the main gameplay area

QuickGig has an introduction screen to teach the mechanics of the game to the player. In order to have higher interactivity throughout the game to improve the game's appeal, the food is draggable. We also implemented an interactive slider in order to add more interactivity to the game. The game is focused on interactivity, but is also made for the player to focus on the number to realize that there is a faster way of completing the orders. The tutorial level makes it easier for the player to realize what they need to do. The success screen in Figure 11 below gives positive reinforcement that the user correctly completed the task whereas the failure screen in Figure 12 below tells the player they need to reevaluate the steps they made during the game.

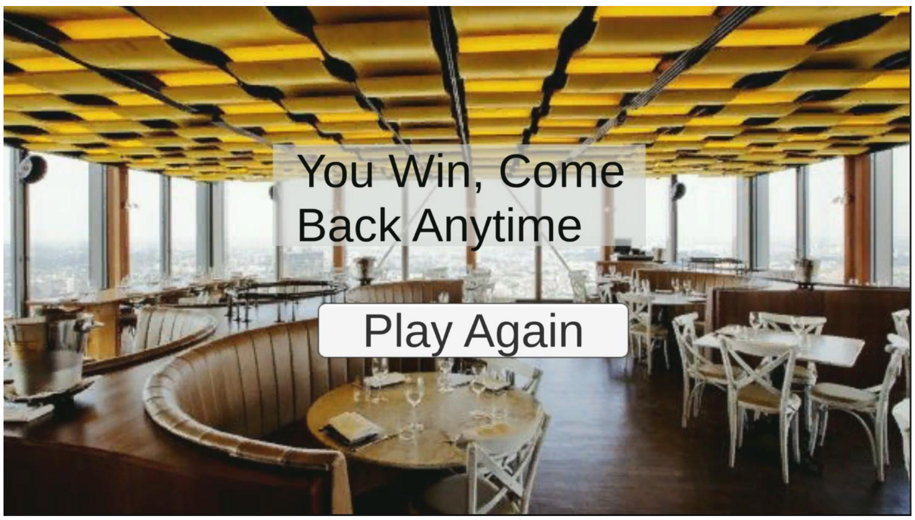


Figure 11: *Quick Gig's* victory screen

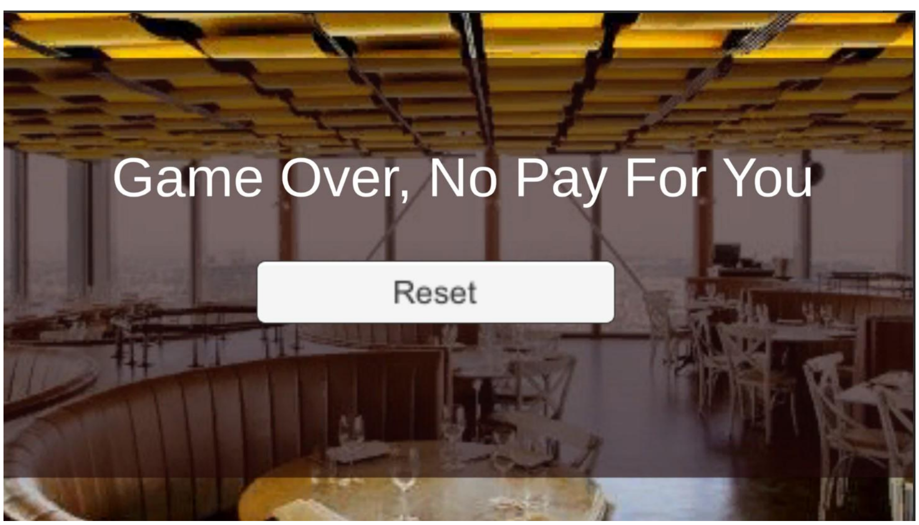


Figure 12: *Quick Gig's* game over screen

3.5: Minigame: *Lightning Change* Methodology

In the game, *Lightning Change*, the player is a cashier, in which they must use basic mathematics skills to calculate the amount of change for customers. These basic math skills include addition and subtraction. The game starts by showing a client wanting to purchase an item. The price of the item and the client's payment are then shown. Then, the player is tasked with calculating the difference and offering the client the correct amount of change. This is done by dragging the required number of bills from the register into the indicated tray. The objective of the game is to complete as many transactions as possible within a given time limit.

First, we created a paper prototype that would enable us to visualize the game's aesthetics, as seen below in Figure 13. After a few iterations of the paper prototype, we were able to settle on a general layout for the interactable elements, the basic gameplay loop, and other aesthetic elements such as an animated face on the cash register reacting to the game state.

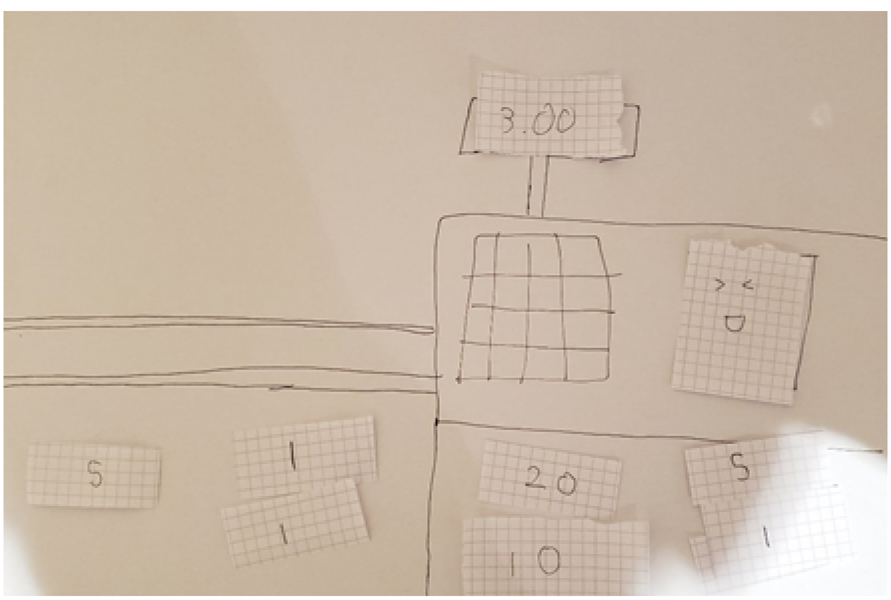


Figure 13: Paper prototype for the minigame *Lightning Change*

Once all the basic elements of the game were decided, we proceeded to implement them in Unity, as seen below in Figure 14. At this point, the game currency's drag-and-drop functionality was affected, which caused the game to not detect when the bills were clicked. The system in charge of detecting whether the bill had been released in the change tray or not was also found to be faulty. After two weeks of effort, we managed to replicate what we had made on paper. We then added a score, which is incremented by one point each time the player is able to provide the correct change. Next we implemented a timer and a lives system. Each time the player is presented with a transaction, a timer begins to count down. If the player is unable to provide the correct change within the time limit, they are not awarded a point for their score, and instead lose one of three lives, which track the player's failures. The game ends once the player runs out of lives, with their final score being equal to however many points they obtained prior.

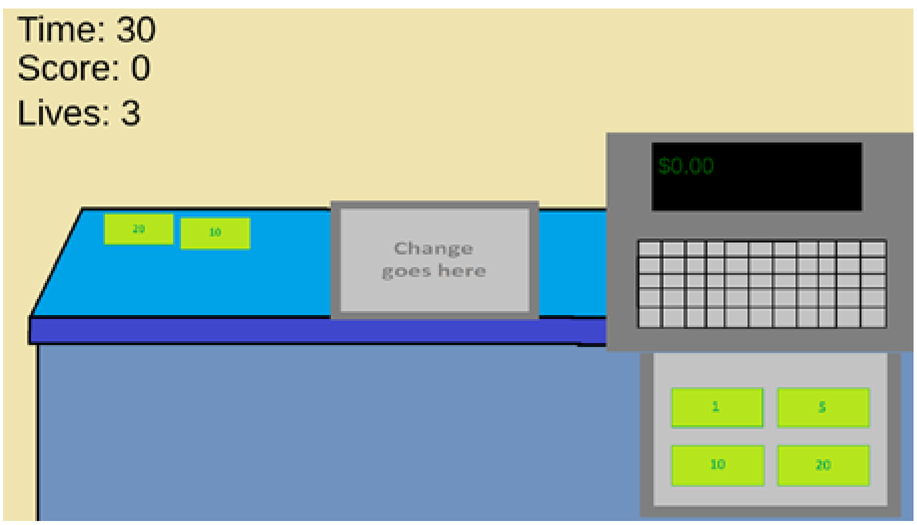


Figure 14: Early version of *Lightning Change* with placeholder art

Afterwards, more adjustments were needed. Testing revealed that the game's visual elements displayed at different sizes when played on different machines, resulting in inconsistent or even disruptive scenes. We then rebuilt the game scene from scratch, which was necessary to introduce the system that would ensure consistent visuals even on different machines. This seemed like a simple task, since the game functioned the same way and no code would need to be changed. Unfortunately, this change broke the bills' drag-and-drop functionality once more, and additional time was spent fixing those issues.

3.6: Main Menu Methodology

We implemented a main menu that combines all of the minigames within *WPI Part-Time Hero*, as seen below in Figure 15. This allows users to navigate between the different minigames.

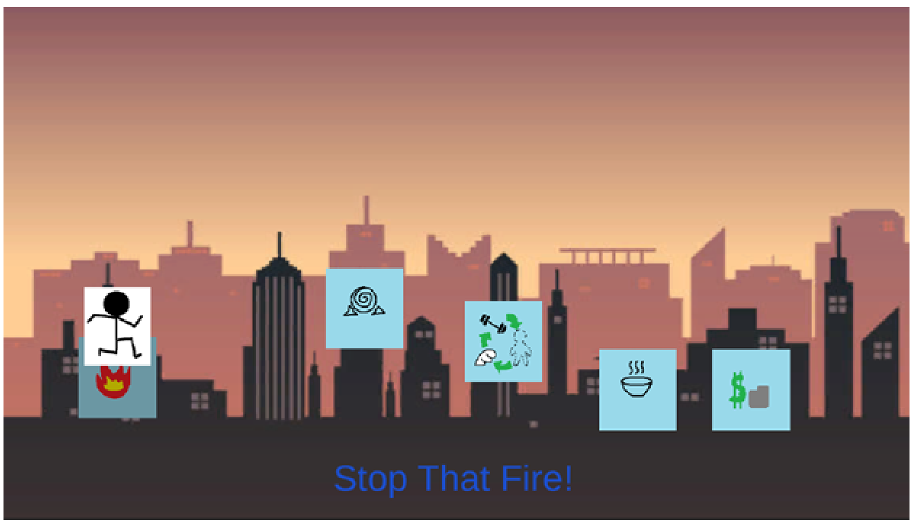


Figure 15: Early version of the game's hub area with placeholder art

Chapter 4: Testing, Results and Conclusions

We had two major testing inquiries over the course of this project. The first testing inquiry was completed during a WPI event in November 2020 known as AlphaFest. This inquiry was primarily focused on examining the functionality of our game. After this testing was completed, we were able to make edits to increase the usability of our game, as explained in Chapter 3. The second testing inquiry was completed over the course of January-March 2021. This testing was focused on the effectiveness and appeal of our game towards our target audience. We analyzed this testing to investigate the amount of interest regarding math and computer science that our game peaked in participants.

4.1: AlphaFest Testing

On November 20, 2020, the game was played in AlphaFest by WPI students. AlphaFest is an event held once a year and is used to test games that students have been developing. All IMGD students who are making a game for their MQP must participate at AlphaFest, others who have made a game can join if they want to.

WPI Part-Time Hero was tested with a semistructured interview as the participants played the game. Our tests showed us that people who did not have any knowledge about computer science or engineering had no clue how to progress through the game. One example is with the *Fortune Teller* minigame where the truth and false statements were confusing to players with limited STEM knowledge. Also the terminology for *Stop That Fire* confused non CS majors who did not understand what a String or Boolean was. All the minigames were also criticized for not having a clear goal or theme when playing. Our tests were successful in figuring out what needed to be improved in our games. This helped us fix problems with our games as we moved forward with development.

From the testing, it was apparent that many people were confused on how to play the games and what the goal of the games were. We implemented some of the minigames with helpful hints and tutorial pages which could be accessed at any time, this made the games easier to follow while playing. Another way that we changed our minigames was by adding sound and visual cues to signify to the player that they made a correct or incorrect action. We knew that if *WPI Part-Time Hero* was confusing to undergraduate students, the game would be impossible for children, who are our target audience. Although the goal of *WPI Part-Time Hero* is to introduce children to CS concepts, the games should be easy enough for children to understand their goal but challenging enough for our audience to have a fun time. Alphafest brought forth many problems that we did not notice until the event. Such as the art style, which We were able to address these problems more efficiently due to the feedback we got from the notes we had written down.

4.2: Project Testing Goals

The goal of this project is to create an effective learning tool for children that is also an engaging game to play. To ensure that both goals were accomplished, we arranged for testing sessions to put the effectiveness and appeal of the game to the test. The format of our testing is a pre-post survey. A pre-post survey is a measurement used before and after an event to help gauge what changed during testing. In our case, the event we're testing is the effectiveness and appeal of our game. The reason we used pre-post surveys is because they are an effective way to test concepts within a study over a short period of testing time (Davis, 2002). We designed the pre-post survey and the testing experience to be short in order to keep the subject's attention.

In order to achieve our goals of testing for effectiveness and appeal, we focused on testing participants for their interest in playing our game, as well as testing participants for their interest in learning topics in mathematics and computer science. Our testing process involved having participants fill out a five-minute survey before interacting with our game regarding questions related to interest in mathematics/computing, as well as efficacy in those subjects. Then, participants interacted with our game for ten minutes. Afterwards, participants filled out another five-minute survey regarding interest in mathematics/computing. In regards to the appeal of our game, participants commented on how enjoyable their testing experience was.

4.3: Outreach Efforts

After AlphaFest testing in November 2020, we began focusing on testing our games for the target audience: students in the 2nd and 3rd grade (ages 7-9). In February 2021, we began extending our research out to local elementary schools and after-school programs in the Worcester area. A list of these contacts can be found in Appendix A. We attempted to reach out to these contacts by designing a flyer and writing an outreach email template that contained information regarding the components of our study. The flyer can be found in Appendix B. The outreach email template can be found in Appendix C. Unfortunately, we did not receive any replies to the emails and calls that we made to reach out to these contacts.

If we were able to establish a line of communication with any of these schools or programs, the next step would have been to organize a Zoom conference call with the participants, at least one adult guardian, and at least two researchers within our team. Before the meeting, we would ensure that the participants have access to the pre-post survey that was sent to them, as well as ensuring that our game was successfully uploaded onto the participant's local machine. During the Zoom conference call, the participant would spend the first five minutes filling out the first part of the pre-post survey. Then, we would instruct the participant to play *WPI Part-Time Hero* for about ten minutes. Next, the participant would spend about five minutes filling out the last part of the pre-post survey. Finally, we would thank the participant and their adult guardian for their time, and end the Zoom conference call. As stated above, we were unable to establish contact with any of the schools or organizations that we reached out to, so we were unable to gather actual participant data in this context.

4.4: Results and Conclusions

4.4.1: General Outcome

During the development of any product, testing is required to ensure that it accomplishes its purpose. Every entity has their own process for conducting tests. Our protocol included creating a prototype build of our game, a pre-post survey, a flyer, and an informed consent agreement, as explained in our outreach efforts. Once we had all of the following, it needed to be approved by the WPI IRB (Institutional Review Board) to ensure that the tests we conducted were safe to the participants. After the approval, we began searching for testers by sending out flyers. They were sent to professors at WPI (Worcester Polytechnic Institute) and groups in Worcester. All communication had to be done virtually due to the Coronavirus pandemic, which prevented us from meeting with anyone in person. In general, communicating with potential testers is the most time consuming part of the testing process, so we made sure to do it immediately. Next, we needed to polish our prototype. This included adding in our artist's assets, merging all of our minigames into one, and testing for errors. By the time we finished our edits, we finally began hearing back from our contacts. Within our timeframe, we were only able to find nine participants.

Nine participants was not a large enough sample size to make general claims about the hypotheses we were testing. Therefore, we will not be able to conclude anything definitively based on the data we have. Some of the main aspects we will be focusing on from our testing results include the impact of active learning and how our game design impacted different gender's interest in math and computer science. The goal of our project is to create an effective learning tool for children that is also an engaging game to play. It is unrealistic to assume that the testers could learn an entire topic in a ten minute period. To accommodate for this, we chose to measure if their interest increased in computer science and math after interacting with our game.

This was done by asking the participants in the pre-post survey if they like math and computers before and after playing our game. We also focus on the gender of the participant to see if females' interest in math and computer science increased after playing our game. As discussed prior, one of our project goals is to encourage girls to increase their interest in computer science and mathematics.

4.4.2: Active vs. Traditional Learning

We analyzed our survey results to see if our active learning teaching method with the game increased the players interest in math and computer science. The participants answered questions based on how much they liked math and using computers in a pre-test survey before playing the game. Then, they answered questions on whether they were more interested in computers and math with a post-test survey. We recorded changes in the participants interest in: math, computers, and how computers work.

The question that asked if the user liked math in the pre-test was compared to the user's answer to if the game increased their interest in math in the post-test. Out of the four participants who did not like math, three of them were more interested on the subject after playing the game. Out of the five participants who liked math at the beginning of testing, four of them had an increase of interest in the subject after playing our game. Three participants who were not interested in math were more intrigued on the topic after playing the game. We cannot make any general claims due to the small sample size, but we can infer that our game had a positive impact on the participants' interest towards math.

The next question that was asked was does the user like computers in the pre-test and was compared to the user's answer to if the game increased their interest in computers in the post-test. Out of the three participants who did not like computers, only one of them was more interested in the topic after playing the game. Out of the six participants who liked computers in the pre-test, five had an increase of interest in the subject after testing. One participant who was not interested in computers was more intrigued on the topic after playing the game. There was little change in the users' interest in computers in general. The game did not make an impact on our participants in that topic.

The same question from before was asked to see if the participant liked computers in the pre-test and was compared to the answer to if the game increased their interest in how computers work in the post-test. Out of the three participants who did not like computers, two of them were more interested in the topic after playing the game. Out of the six participants who liked computers in the pre-test, five of them had an increase of interest in the subject after testing. Two participants who were not interested in how computers work were more intrigued on the topic after playing the game. There was a small increase in users interested in how computers work, but more players need to be tested before making any general claims regarding how effective our game is in this topic.

Due to having only nine participants take part in our study, we cannot make any definitive conclusions. However, from the data we did collect, there was an overall increased interest in math and how computers work. The increase from the interest of math as a result of our game shows that active learning had some sort of effect on the participants. Nearly all the participants recorded comments in the post-survey describing that they were entertained with *WPI Part-Time Hero*. Three participants recorded in the post-test that they did not learn anything new from playing our game. This shows that educational games can be entertaining and teach players new concepts of learning. However, it's important to note that more testing needs to be done in order to find the benefits of active learning versus traditional learning.

4.4.3: Gender Biases

In addition to studying active versus traditional learning, we also analyzed our survey results for biases in gender and how individual's gender might have affected their experience when playing the game. Out of our nine respondents, we had five girls and four boys. First, we will discuss the data from the boys' and girls' answers. Then, we will discuss what the results may mean.

Out of the participants who responded that they identify as a boy, all four of them responded that they enjoy using computers. Only two of them had any experience with programming. The same two boys also said that they enjoy math, while the other two did not enjoy math. However, after they had played our game, all four boys reported that they were more interested in math, computers, and how computers work. In addition, the same two boys who did not have programming experience recorded in the post-survey that they would play the game again.

Out of the participants who responded that they identify as a girl, results were more varied. Only two of the girls reported that they liked using computers, despite two of them reporting that they use computers for six or more hours each day. The two girls who did record that they enjoy using computers responded that they did not enjoy math. Only one girl had any programming experience before our testing. After playing our game, two of the five girls said that they were more interested in computers. Lastly, three of the five girls responded that they would not play our game again.

We cannot make direct conclusions from our small sample size of only nine respondents. However, from our small study, it seems that boys tend to enjoy our game more than girls do and would play it again, whereas girls would not. Boys also seemed to be more interested in mathematics and computers after playing our game than girls based on our findings. All respondents said that they are more interested in video games after playing our game, despite less than half claiming that they enjoyed our game. This could possibly indicate that *WPI Part-Time Hero* has the potential to retain players' interest. Therefore, our game has the capability to keep users engaged in learning new topics for a prolonged period of time.

Chapter 5: Postmortem and Future Works

5.1: What Went Right

5.1.1: Game Design Process

We did not have to rework our game design concept because we started off with a design that would work under many situations. Collections of games or minigames have been used in video game arcades and home systems alike. Due to the variety in them, they may appeal to many different types of people. A game that serves a similar purpose for a similar audience would be the game *Logical Journey of the Zoombinis* (1996). This game is a set of 12 puzzles that aim to teach logical thinking skills through their gameplay. Our games use a similar model for teaching children computer science concepts such as Boolean logic and data types. *Logical Journey of the Zoombinis* won numerous awards for best educational game in 1996. We followed these previous game models throughout our design process. Thus, our development time was more efficient because our game design was clear.

5.1.2: The Artwork

With the help of our artist, Christian O. Adler, we managed to establish an art style that would appeal to 2nd and 3rd graders and would be easy for other developers to mimic. In particular, we decided to go for a simplistic cartoon art style. It is important to have a simple style because it allows for future work to easily expand upon our game. In the case that a future project does not have a dedicated artist, it is imperative that art created later can still match our current design. In addition, the artwork created looks visually appealing and the style is very common among kids of the seven to nine age range. The art style is entertaining for children.

5.1.3: AlphaFest Testing

AlphaFest is a way for WPI students, typically Interactive Media and Game Development majors, to get their game tested by other WPI students. Since the project we developed was a game, AlphaFest was a good option for having other WPI students test our project. AlphaFest is intentionally held in November so that students can begin testing their game early. It was a great source of advice for each of the minigames we created, such as if the games were too easy. It also helped us learn about bugs in the minigames we had created. Many of the minigames included screen resolution issues, but there were issues unrelated to screen resolution that needed to be fixed as well. An example of another bug that we encountered was that buttons were not created in the correct position for *Future Teller*. AlphaFest aided us in refining how each minigame should play and helped us discern what features were entertaining and which needed improvement. Although some of the games were in a better state than others, the feedback was essential to providing us with information and ideas we might not have considered otherwise.

5.2: What Went Wrong

5.2.1: Scoping

When a video game begins development, creators tend to be ambitious and hopeful in regards to their final outcome. The time it takes to make a product is often overlooked. One of the major obstacles was that our scope was too large. The scope refers to all of the objectives and requirements needed to complete a project. We had planned for many features and minigames to be in the actual game that never made it to the final project. For example, we were planning to create more than six minigames. We began by brainstorming at least two minigames during the planning stage of our project per developer. The plan was to have each developer create two minigames that they envisioned. For example, one member developed *Stop That Fire*, but they also were planning to create a minigame in which the player sorts animals in a zoo. This would be used to teach children how to sort items in different list types including arrays, lists, and stacks. Unfortunately, there was not enough time to develop these extra minigames.

Another idea we had to remove from the final project was a customizable player character. We envisioned that each player would have a more unique experience with the game if they had a distinct character. The character would additionally appear in each of the minigames and give the player hints if they encountered any obstacles. Unfortunately, we overestimated the scope of the project, and could not implement this feature.

Our final idea that we could not implement was a hub world. A hub world refers to a location in a video game in which all levels are accessed. A menu serves the same purpose, but the hub world is meant to keep the player engaged in the game. Our idea for a hub world would be similar to that of *Club Penguin* (2005). The player would use their customizable character and click on where they want that player to walk. There would be different buildings that represented each of the minigames, and the player could click on them to access that minigame. Similar to our previous points, we were unable to add this due to overestimating the scope of the project. Instead, we opted for an interactive menu. The player moves a character to a building that represents each level. It is a simplistic design, but it accomplishes our goal of making the menu more enjoyable for the player.

5.2.2: Technical Difficulties in a Virtual Environment

During the years of 2020 and 2021, the world was faced with the Coronavirus pandemic. Due to this public health concern, our team members had to socially distance and conduct all meetings virtually. This meant any communication our group had was completed over Zoom video calls, emails, and the GroupMe messaging platform. This led to many unforeseen issues, leading us to make adaptations to our project.

A complication was exchanging files over the internet. The easiest way to exchange files is to meet in-person and do integrations together over GitHub or using a flash drive. We did not end up using GitHub often. The only way we were able to transfer files to each other was through emailing packages. Unity allows users to put an entire game file into a package so it can easily be shared with someone. Due to the social distancing constraints we described, integration became more difficult than expected.

Throughout our development, we found that it is difficult to debug games over a virtual environment. Each developer was responsible for implementing one minigame within *WPI Part-Time Hero*. We used Zoom video calls to assist each other in debugging our minigames. In this environment, users can share their screen with the rest of the members of that specific video call. However, one user cannot control another member's screen. Therefore, we could only assist each other through verbal instructions. This process is incredibly time consuming, and we spent much of our development time attempting to assist each other.

Testing with participants in a virtual environment became an unforeseen hurdle. As mentioned before, it is difficult to transfer files to people, and transferring an entire game to someone is just as difficult. This process was even more complicated on Macintosh computers. Macintosh computers are unable to open zipped files. Therefore, we had to transfer the whole game over onto a Macintosh machine using a flash drive. Even when the game files were sent to our participants, they still had difficulty installing the game on their local machine.

5.2.3: Integration

One major issue that we had with this project was integration. Integration is the process of combining different pieces of a project together. Typically, computer science students use the GitHub platform to take care of integration. The only problem with using GitHub is that it can lead to merge conflicts. A merge conflict occurs when two people simultaneously edit the same file. Merge conflicts can be time-consuming to resolve. In addition to this complication, we had difficulty using GitHub within Unity to properly make edits within our program. Therefore, we manually integrated our code. One person in the group imported everyone else's minigames together on one machine.

5.3: Future Work

When developing the game, we originally considered making each of the minigames accessible from a map, in which characters can walk to each minigame. This map would offer players a small area to freely move a character to interact with their surroundings. In this area, players could then move to interactable elements to begin each of the minigames. This system would work in contrast to the game's current main menu, which offers a strict list of options presented in a stylized manner.

We also intended for users to have a player avatar, whose appearance could then be freely customized with various accessories. More additions for their character's appearance could be obtained by spending time with the game and with good performance in each of the minigames. An important consideration would be the gender of the avatar, which should preferably be neutral in order to promote equality among genders and encourage girls to play our game. These changes would then be reflected within the minigames by having the player character's outfit change accordingly. Due to limitations in art acquisition and lack of time, we switched to having a fixed player character. This significantly simplified development since we didn't need to implement the underlying character modification systems or create art for each potential accessory.

In order to broaden the educational topics covered in *WPI Part-Time Hero*, more minigames could be added to our game that cover additional concepts. For example, one topic could be sorting and ordering, which is not clearly accounted for in any of our current minigames. Another example is a minigame focused on arrays and lists. With additional resources, time could be spent designing a minigame dedicated to these topics, as well as other aspects of computer science.

Works Cited

2018 NSSME+. NSSME. http://horizon-research.com/NSSME/2018-nssme.

Alex FriedmanFounder at Brooklyn Math TutorsAlex has enjoyed tutoring math. (2018, October 4). *The Long-Term Effects of Ineffective Teachers*. Brooklyn Math Tutors. https://www.brooklynmathtutors.com/the-long-term-effects-of-ineffective-teachers.

All Winners: Kids' Choice Awards: Nickelodeon. Kids' Choice Awards | Nickelodeon. https://www.nick.com/kids-choice-awards/all-winners.

Bachelor, R. (2012, April 30). Exploring the effects of active learning on Retaining essential concepts in secondary and junior High Classrooms. Retrieved March 15, 2021, from https://eric.ed.gov/?id=ED531546

Bieg M, Goetz T, Wolter I and Hall NC (2015) Gender stereotype endorsement differentially predicts girls' and boys' trait-state discrepancy in math anxiety. *Front. Psychol*. 6:1404. doi: 10.3389/fpsyg.2015.01404

Comiskey, D., & McCartan, K. (2011). Video: An Effective Teaching Aid? An ArchitecturalTechnologist's Perspective. CEBE Transactions, 8(1), 25-40.

Computer Science in California Schools: An Analysis of Access, Enrollment and Equity. Kapor Center. https://www.kaporcenter.org/csinschools/.

Davis, G. A. (2002). Using a Retrospective Pre-Post Questionnaire To Determine Program Impact.

Dweck, C. S. (2008). *Mindset: The new psychology of success*. Random House Digital, Inc..

Dweck, C. S., & Yeager, D. S. (2019). Mindsets: A view from two eras. *Perspectives on Psychological science*, *14*(3), 481-496.

Earlychildhood NEWS - Article Reading Center. http://www.earlychildhoodnews.com/earlychildhood/article_view.aspx?ArticleID=119.

Erturan, S., Jansen, B. An investigation of boys' and girls' emotional experience of math, their math performance, and the relation between these variables. *Eur J Psychol Educ* 30, 421–435 (2015). https://doi.org/10.1007/s10212-015-0248-7

Freeman, S., Eddy, S., McDonough, M., Smith, M., Okoroafor, N., Jordt, H., & Wenderoth, M. (2014, June 10). Active learning increases student performance in science, engineering, and mathematics. Retrieved March 15, 2021, from https://www.pnas.org/content/111/23/8410

Geri, N., Winer, A., & Zaks, B. (2017). Challenging the six-minute myth of online video lectures: Can interactivity expand the attention span of learners? Retrieved March 15, 2020, from http://www.iiakm.org/ojakm/articles/2017/volume5_1/OJAKM_Volume5_1pp101-111.pdf

Guardia, J. J., Del Olmo, J. L., Roa, I., & Berlanga, V. (2019). Innovation in the teaching-learning process: the case of Kahoot!. On the horizon.

Grant, A., & Anthony Grant (41 Articles Published) . (2019, December 20). *Why You Shouldn't Learn to Code With Codecademy*. MUO. https://www.makeuseof.com/tag/4-reasons-shouldnt-learn-code-codeacademy/.

Halpern, Diane F., Joshua Aronson, Nona Reimer, Sandra Simpkins, Jon R. Star, and Kathryn Wentzel. 2007. Encouraging girls in math and science (NCER 2007-2003. Washington, DC: National Center for Education Research, Institute of Education Science, U.S. Department of Education.

Huang, X., Zhang, J. & Hudson, L. Impact of math self-efficacy, math anxiety, and growth mindset on math and science career interest for middle school students: the gender moderating effect. *Eur J Psychol Educ* 34, 621–640 (2019). https://doi.org/10.1007/s10212-018-0403-z

Introducing the Root® robots for coding, discovery, and play. iRobot. https://www.irobot.com/Root.

Jacksepticeye (Director). (2015, October 23). *BRINGING home the Bacon | human Resource Machine* [Video file]. Retrieved March 14, 2021, from https://www.youtube.com/watch?v=5fvpSpQeRWc&t=773s

Jones, S. (2019, January 8). *STEM Instruction: How Much There Is and Who Gets It*. Education Week. https://www.edweek.org/teaching-learning/stem-instruction-how-much-there-is-and-who-gets-it/2019/01#.

LEGO® MINDSTORMS® About: Official LEGO® Shop US. Home. https://www.lego.com/en-us/themes/mindstorms/about.

Main, S., & O'Rourke, J. (2011, March). 'New directions for Traditional Lessons': Can handheld game CONSOLES enhance mental MATHEMATICS SKILLS? Retrieved 2020, from https://ro.ecu.edu.au/ajte/vol36/iss2/4/

Short, D. (2012). Teaching scientific concepts using a virtual world-Minecraft. Teaching science, 58(3), 55.

Weinschenk, S. (2020). *100 Things every designer needs to know about people*. Peachpit Press/New Riders.

Wikimedia Foundation. (2020, November 30). *List of educational video games*. Wikipedia.

https://en.wikipedia.org/wiki/List_of_educational_video_games.

Appendices

Appendix A: Preliminary Outreach List to Find Participants for Testing

School/Program Name	**Website**	**Contact Info**
Elm Park Community School	https://worcesterschools.org/school/elm-park-community-school/	Principal: Lucas Donohue Email: DonohueL@worcesterschools.net School Hours: 7:55 am - 2:00 pm Phone: 508-799-3568 Fax: 508-799-8216
Francis J. McGrath Elementary School	https://worcesterschools.org/school/francis-j-mcgrath-elementary-school/	Principal: Ellen Moynihan Email: moynihane@worcesterschools.net School Hours: 8:25 am - 2:30pm Phone: 508-799-3584 Fax: 508-799-8235
Boys & Girls Club of Worcester	https://www.bgcworcester.org	65 Boys & Girls Club Way, Worcester, MA 01610-2520 508-754-2686 · fax 508-754-7635 · info@bgcworcester.org
YWCA of Central Massachusetts	https://ywcacm.org/	https://ywcacm.org/contact/
YMCA of Central Massachusetts: Central Community Branch	http://www.ymcaofcm.org/locations/central-community-branch/	https://www.ymcaofcm.org/contact/

Appendix B: Outreach Flyer

Appendix C: Outreach Email

Hello {insert name of correspondent}!

We are a team of five computer science undergraduate students from Worcester Polytechnic Institute. We are working on a project that focuses on educating children (ages 7-9) on computer science concepts through the use of educational video games. We have been researching math/computer efficacy, anxiety and interest in math/computer concepts among 2nd-3rd graders.

We are reaching out to you because we would like to have 2nd and 3rd graders play our game for testing purposes. All meetings will be conducted over Zoom video conference. At the beginning of a testing session, participants will spend ~5 minutes taking a pre-game survey asking questions regarding math/computer efficacy, which you can view here. Next, participants will spend ~10 minutes playing our game. You can view instructions on how to download and play our game here. The game consists of 5 minigames that are designed to teach the following concepts:

- Basic boolean logic
- Identifying different data types
- Basic loop concepts for programming
- Basic mathematics skills involving addition and subtraction

Finally, participants will spend ~5 minutes taking a post-game survey, which you can view here. A more detailed description of how these meetings would work can be found in this document. Please view this flyer for more information as well. We are interested in learning about self-efficacy, anxiety, and interest in math/computers amongst 2nd-3rd graders. We hope this game is fun, and fits into your current curriculum.

We’d be happy to follow-up with a phone conversation or a Zoom video call. You can also contact our supervising faculty, Professor Therese Smith (tmsmith@wpi.edu) and Professor Jennifer deWinter (jdewinter@wpi.edu) if you have any further questions. If we don’t hear from you by {insert date here}, we will be following up with a check-in email to see if you are interested in participating. Thank you so much for taking the time to read this email, and we look forward to hearing back from you soon!

Sincerely,
Alexander Bell, Haley Hauptfeld, Javier Marcos, Oliver Rayner, Linda Puzey

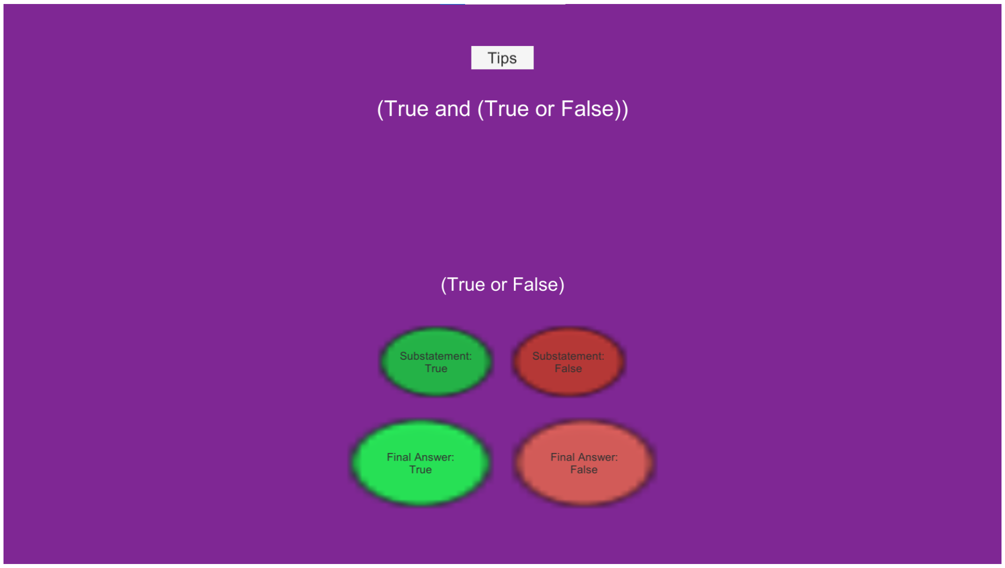
Appendix D: Protocol for Hosting Playtesting

Educational Games for Teaching Children Computer Science Study:

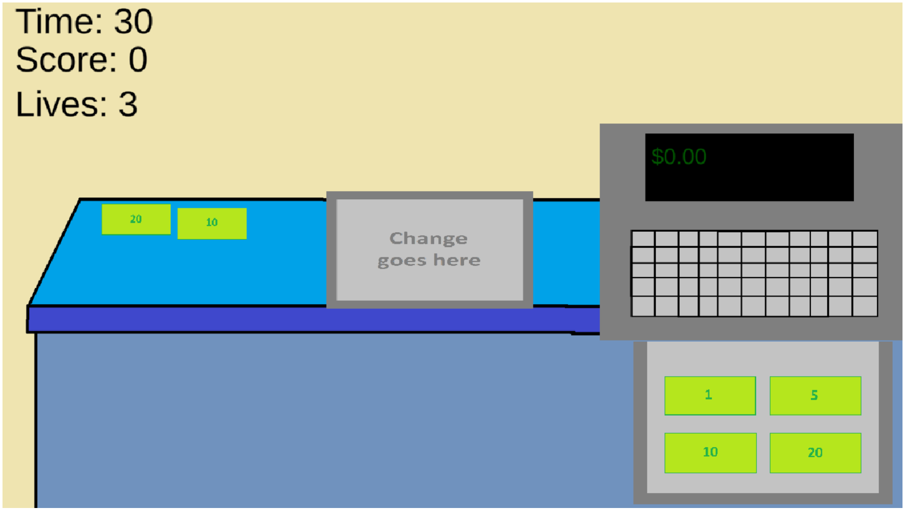
The purpose of our study is to create a video game that will help children (ages 7-9) to learn basic computer science concepts. To accomplish this, we will test the effectiveness of our educational video games with children to ensure that the game is an entertaining, effective learning tool for our target audience. Participants will be recruited by contacting schools in the Worcester area, introducing ourselves to a principal or teacher through email, introducing them to the research and asking them if we could meet over a Zoom video conference with student participants in the 2nd and 3rd grade (ages 7-9) and their teacher and/or adult guardian. A template/draft of this email can be found here. A flyer is also included in the email to attract participants.

The video game will be free to download on Google Drive, and will not require an account to be downloaded. The game is a collection of 5 minigames, each covering different computer science concepts, including loops, booleans logic, basic math operations and data types. The game's file size (~50MB) and system requirements will be minimal, and may be kept or deleted by the participant if they please. Below are screenshots of each game that a participant could potentially test:

Future Teller:



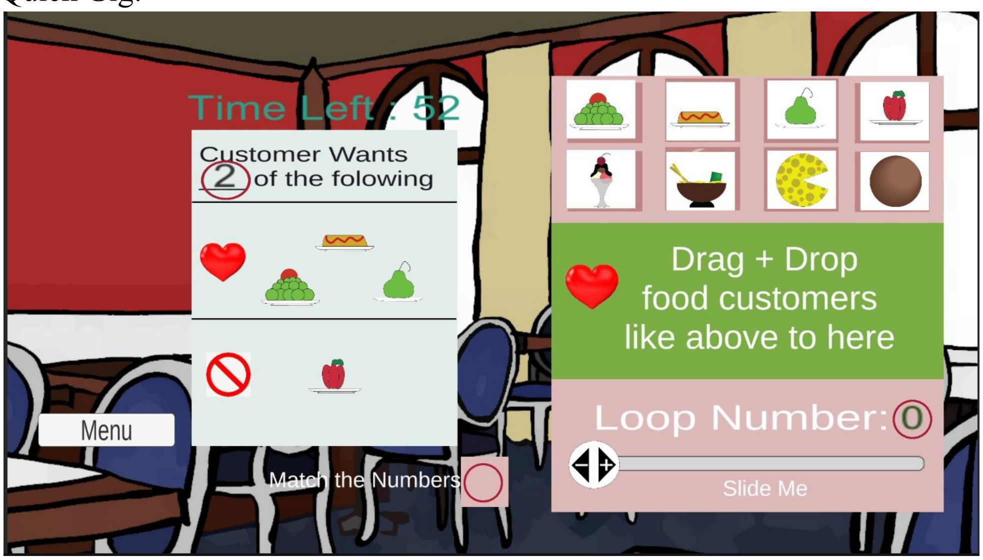
Lightning Change:



Stop That Fire:

Superpower Selector:

Quick Gig:



Once guardians respond to our outreach email, we will explain the procedure in greater detail and answer any questions they may have. After initial outreach, guardians and participants will be sent a consent form (which can be found later in this document) to sign before participating in the study, and researchers will schedule a Zoom video call that will include at least two researchers, at least one guardian, and at least one child participant to test out the minigames described in the previous paragraph. The Zoom video call will last approximately 30 minutes per participant at a time. The occurrence of the Zoom will be scheduled at a day and time that at least two researchers, the participant, and a guardian can all attend the meeting.

The experiment will take place in a scheduled zoom video conference. Participants are greeted by two researchers and everyone will introduce themselves. The child participant and their guardian are then given a brief introduction to the study, the procedure and its purpose while also reminding them that their information will be confidential and that they may withdraw at any time.

During the scheduled Zoom meeting, the participant will then be sent a link to a pre-evaluation survey that lasts roughly 5 minutes and covers loops, booleans logic, basic math operations, and data types. Then they will be sent a link for a free, direct download for the developed game that consists of all 5 minigames that were previously described. The participant will then be asked to play the 5 games included for approximately 3-5 minutes per game.The researchers will be taking notes on the participant's reactions to the games during this time. The participant will then be handed a third link to a post-evaluation survey, similar to the first, in order to test if the participant gained knowledge of basic computer science concepts after playing our game. This is the link to the pre/post game survey.

The participant may move on from each game if they feel ready, as well as skip any of them. While it is preferred for the participant to play all 5 games, some information may be obtained from some of the games played so long as they complete both surveys. Both surveys are a requirement for the purpose of testing the entire game. Afterwards, the results of the surveys will be used to determine the effectiveness of the video game as a tool for teaching basic computer science concepts to children (ages 7-9). After the project ends, this information will be stored in a private Google Drive account.

Appendix E: Informed Consent Form

Informed Consent Agreement for Participation in a Research Study

Investigator: Therese Smith

Contact Information: tmsmith@wpi.edu

Title of Research Study: Games for Teaching Children

Sponsor: Worcester Polytechnic Institute

Introduction: You are being asked for consent for your child to participate in a research study. Before you agree, however, you must be fully informed about the purpose of the study, the procedures to be followed, and any benefits, risks or discomfort that your child may experience as a result of their participation. This form presents information about the study so that you may make a fully informed decision.

Purpose of the study: The purpose of this study is to identify the effectiveness and appeal of an educational video game meant to teach children about various computer science topics including loops, boolean logic, basic math operations, and data types.

Procedures to be followed: Your child will complete a 5-minute pre-evaluation survey on loops, boolean logic, basic math operations, and data types. They will then be asked to download a collection of 5 games and play each one for 5 minutes or until they choose to move on. This is a direct, free download and while playing all the games is preferred, your child may choose to skip any number of them. Finally, they shall take a 5-minute post-evaluation survey, covering the same topics.

Risks to study participants: There are no foreseeable risks associated with this research study.

Benefits to research participants and others: You and your child will have an opportunity to enjoy and comment on a new game under active development meant to help teach children from grades 2 to 3 about various topics related to Computer Science, such as loops, booleans logic, basic math operations, and data types. Information collected will help gauge the game's engagement, as well as its effectiveness as a learning tool for children of its target age

Record keeping and confidentiality: Data collected will not be identifiable and will be completely anonymous. Records of your participation in this study will be held confidential so far as permitted by law. However, the study investigators, the sponsor or it's designee and, under certain circumstances, the Worcester Polytechnic Institute Institutional Review Board (WPI IRB) will be able to inspect and have access to confidential data that identify you by name. Any publication or presentation of the data will not identify you.

Compensation or treatment in the event of injury: There is no foreseeable risk of injury associated with this research study. Nevertheless, you or your child do not give up any of your legal rights by signing this statement.

For more information about this research or about the rights of research participants, or in case of research-related injury, contact the Investigator listed at the top of this form. You may also contact the IRB Manager (Ruth McKeogh, Tel. 508 831-6699, Email: irb@wpi.edu) and the Human Protection Administrator (Gabriel Johnson, Tel. 508-831-4989, Email: gjohnson@wpi.edu).

Your participation in this research is voluntary. Your refusal to participate will not result in any penalty to you or any loss of benefits to which you may otherwise be entitled. You may decide to stop participating in the research at any time without penalty or loss of other benefits. The project investigators retain the right to cancel or postpone the experimental procedures at any time they see fit.

By signing below, you acknowledge that you have been informed about and consent to be a participant in the study described above. Make sure that your questions are answered to your satisfaction before signing. You are entitled to retain a copy of this consent agreement.

__________________________________ Date: ___________________
Participant's Parent Signature

____
Participant's Parent Name (Please print)

__________________________________ Date: ___________________
Study Participant Signature

____
Study Participant Name (Please print)

__________________________________ Date: ___________________
Signature of Person who explained this study

Appendix F: Game Survey

Game Survey

1. Are you a boy or a girl?

Mark only one oval.

- ( ) Boy
- ( ) Girl
- ( ) Other: ______

2. How old are you?

______

3. How long do you use a computer per day?

Mark only one oval.

- ( ) 0-2 Hours
- ( ) 2-4 Hours
- ( ) 4-6 Hours
- ( ) 6+ Hours

4. Do you like using computers?

Mark only one oval.

- ( ) Yes
- ( ) No

5. Do you like math?

Mark only one oval.

- ◯ Yes
- ◯ No

6. Do you have any experience with programming?

Mark only one oval.

- ◯ Yes
- ◯ No

Stop!

At this point, please play the game!

Post-Game Survey

7. Did you enjoy the games?

Mark only one oval.

- Yes
- No

8. What would make the games more fun?

____

9. Did this game increase your interest in math?

Mark only one oval.

- Yes
- No

10. Did this game increase your interest in computers?

Mark only one oval.

- Yes
- No

11. Are you interested in how computers work?

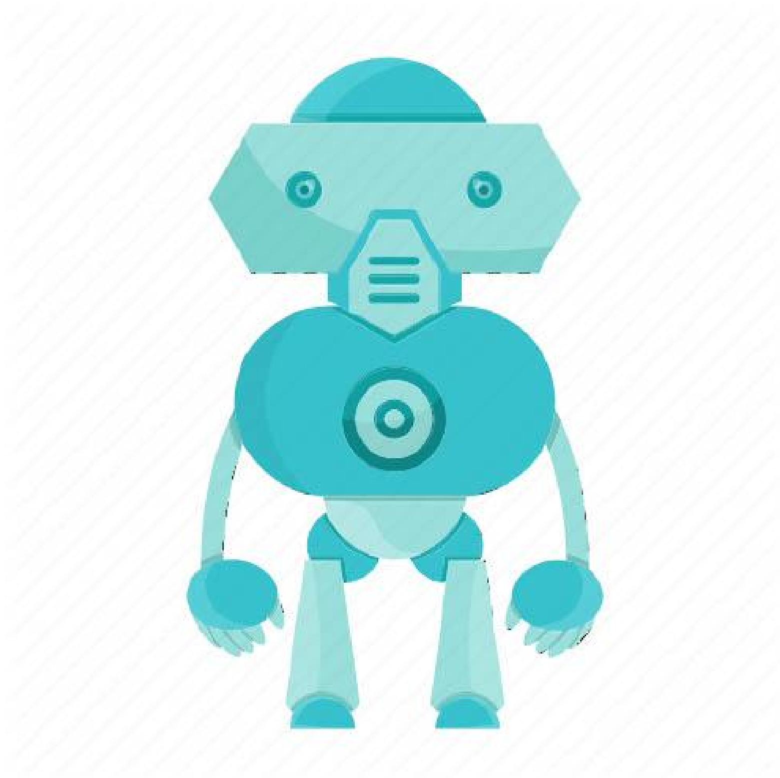
Mark only one oval.

- Yes
- No

12. Are you interested in making games?

Mark only one oval.

- ( ) Yes
- ( ) No



13. Are you more interested in playing games?

Mark only one oval.

- Yes
- No

14. Would you play these games again?

Mark only one oval.

- Yes
- No

15. Which game was your favorite?

____

16. What's one thing you learned?

____

____

____

____

____