Handheld Color Sliding Game

EEL 4914: Senior Design 1

Group Number: Group 24

Group Members:

Josh Bell - Computer Engineering

Eric Espinosa - Electrical Engineering

Linus Fountain - Electrical Engineering

Customers / Sponsors / Significant Contributors:

Matthew Gerber - Computer Science

Wei Sun - Electrical and Computer Engineering

Chapter 1: Executive Summary

As a group we have realized a steady decline in the standalone handheld game market. Over just the past year there has been an 11.5% decline in sales of these types of electronics [1]. This comes as no surprise as most of the target audience for this product may have gone to other game devices such as video game consoles like the Nintendo Switch or maybe a Playstation console. However, as a group we remember from our childhood playing with a vast majority of handheld devices such as the Bop It. Our device is an attempt at attacking this market head on and challenging ourselves to create something that will be well received by a large number of consumers.

Our handheld color sliding game aims to bring this type of game back to life. To do this we hope to have a final product that has a sleek design, sound mechanics and most of all, challenging and rewarding gameplay. To begin this journey we are forming this technical paper that we will use as guidelines and goals that we hope to achieve over the course of this project. We will expand upon this paper as we continue to work towards our final product while simultaneously researching and developing components we will need. After we acquire these components we will have to test our initial design and redesign where it needs to be done. Finally, when we have our final design we will fine tune and make final changes that we believe will provide the best experience for the user.

Chapter 2: Project Information

2.1 Project Motivation

From the start when forming our group, we have made a unanimous decision that we wanted to create an idea that promises enjoyment for consumers, but also ignites our own excitement and passion for something to work on and develop. For all three of us, we grew up in an era of rising technology and games, where there was always something new and developing every week. This is an area that we all enjoy and are familiar with, so we agreed to look into this direction for our project.

We took inspiration from our childhood, and desired to generate our own new and unique game. Our project design is something we believed would be compact, entertaining to use, and comfortable to grip for the consumer. Especially those familiar with more modern day consoles, such as the Nintendo Wii, Nintendo Switch, various arcade machines, and virtual reality(VR) gaming. Furthermore, our design is also user-friendly and comprehensible to individuals of all age groups. We regard this project as a valuable evaluation of our existing skill set, while also serving as a means to expand upon and acquire new abilities.

2.2 Project Description

As a group, we have collectively embraced the concept of a handheld game with a central focus on a mechanical component, as opposed to something like a touchscreen. The system roughly resembles a PlayStation Portable in shape and design, with a central screen and the controls on each side.

When the game commences, eight random colors and directions will be presented to the player on the LCD in a way that visually wraps around the toggle's area of motion. A central color near the toggle corresponds to one of these eight arrows and colors, informing the user of their current target. To advance in the game, the user must manipulate the toggle in the direction past a section that matches the current assigned color, and in the direction of the section's arrow. When the user moves the toggle past the correct arrow zone it will be marked as complete and the section's color will fade away. Then the central color will change the color to the next target. Once all sections have been cleared, the game will restart with less time to complete the new stage. The middle of the display will showcase both the remaining time and the user's current score, which will showcase their progress. The main menu will have the option of seeing all high scores. The system has the option of having the game play on both toggles at the same time or a single toggle that the user can choose.

2.3 Design Sketch of Game Rules

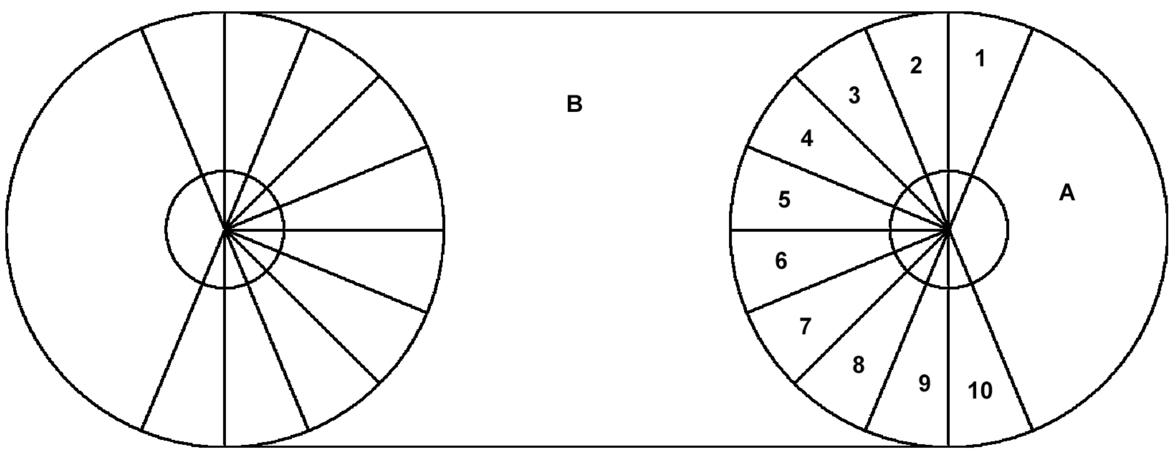

Figure 1 - Render of game rules

Precisely the game rules are as follows: As seen above each toggle has sections one through ten. Visually the player can only see sections two through nine, as one and ten are off screen and only exist within the game's code. Sections two through nine will each be assigned a random color (duplicate colors may occur) and a direction that is either clockwise or counter clockwise. And a color near the center will display what corresponding section may be cleared. For example, if section three is red with a clockwise arrow and the toggles current color is red, the player must move the cursor from section four, to three, to two; and the section three would be cleared. The purpose of section one and ten is so that within the program clearing sections two and nine can be properly detected.

2.4 Project Market Analysis

We have looked at comparable devices that are already in stores such as the game Bop It, Rubiks Revolution, and other games that have similar properties. While looking at these games and their sales history we have found that similar type games have accounted for 4 billion dollars worth of sales over the course of 2022 which is identical to the previous years sales as well as up half a billion dollars from 2020 [1]. We believe that there is plenty of room for a competing product to join the market as the type of product we are designing only takes up about 7.3% of the current toy market, leaving only room for growth with the insertion of a new product. Compared to the Bop It that is currently selling between $12 - $44 and the Rubiks Revolution that is currently selling at $20-$22 we think that hypothetically we would be able to sell our product for $35. While this is greatly below our estimated total cost of production we think that if we were to mass produce this there would be much room to bring the total cost of production down well below the $35 price point.

2.5 Project Goals

Our primary goal for this project is to create and design a completely functional, handheld, mechanical game that promises entertainment and is easy to use for a wide spectrum of consumers. We want this device to have a screen consisting of eight sections on each side that will be used for the gameplay itself. On these eight sections we will have colored arrows, each arrow will light up a specific color. For this to be a stimulating and fun experience to the consumer, we would like to have a central toggle with set zones for each arrow to be able to determine the path the user goes. We would also like to have a speaker that will play a timer of sorts, so the user knows how much time they have left for each level, and some feedback music that plays once you pass an arrow. Once the user fails, we will have a system to keep track of the high score to provide a challenge for the user, so they can keep pushing themselves to beat their score, or invite a friend to compete with them. We would also like to have the main screen have a scrolling text system that would allow us to present messages such as “game over” or “new high score!”.

2.5.1 Basic Goals

- LCD Screen to Present Game and Game info
- Device Keeps Track of Time and Score
- User Interface
- Thumbstick Control
- Speakers to play audio for timer and sound effects

2.5.2 Advanced Goals

- 3.5 mm headphone jack
- Difficulty options

2.5.3 Stretch Goals

- Scrolling text across the main screen to show high scores ect.
- Bluetooth capability
- Phone connectivity
- Highscore Leaderboard

2.5.4 Functional Requirements

The functional requirements this game project must have is being able to start and pause the game. The LCD should provide visual feedback by showing the current color target closer to the toggle, and the targets to hit on the main board. The middle of the LCD screen should display game related information, such as the highest score, a timer, and pause/start, and scores should be updated in real time.

The device has to be user friendly, that is comfortable and easy to use. It should have a functioning toggle joystick that operates for our intended use to control the game and menu. It should be responsive and provide feedback to the user. The game should have a difficulty or progression system. Where from the start you can pick your difficulty, or if level based, then when you beat level 1, you go to 2, etc. This way the game can keep being enjoyed, while also challenging the user to progress and get better.

A nice requirement to include is some type of feedback, whether it be physical, audio or visual. The feedback can include the thumbstick or controller vibrating to let you know when you have passed a color. An audio that plays when you pass a color, or beat a level. Or visual, if we do an LCD screen we can display text on the screen to inform the user of certain actions.

2.5.5 Technical Requirements

We have chosen to use an LCD screen rather than the originally planned LEDs, it will be a small 4-5 inch x 1-2 inch screen. The LCD would display the arrows going left and right, as well as give the arrows the color needed for the game. It would also display a type of cursor that moves in the direction you are choosing, to let the user know which direction they are going so they do not get lost. The cursor is updated in real time, without delay to ensure a smooth experience for the user.

The main menu LCD screen will display the menu at the start of the game, which will have a start option to start the game, a high score option to check what the highest score is on the device, and maybe a volume option to control the volume. When the game starts, the screen will display the current score of the game, and a timer that counts down to showcase how much time the user has to complete it.

For the microcontroller unit, we were deciding between using something like the MSP430FR6989 or an arduino kit, since we are most familiar with C, C++ and Java. The device will be powered by 2 AA batteries, but a rechargeable power source is being considered in order to avoid electronic waste of having to replace the batteries every so often. It will have built in speakers and will either have a switch or button to power the device on, and it might have a volume slider depending on how we decide to control the volume, whether it is physical or digital.

The device might have an audio jack or bluetooth option, but it is not really needed so these features would be added towards the end once we completed our main goals and objectives. The controller would use non-volatile memory, in the form of an electrically erasable programmable read-only memory(EEPROM) or non-volatile RAM(NVRAM) in order to store score data and high score information. This type of memory is used in arcade games and pinball machines, as it is essential for maintaining records and will save the data even after turning off.

2.6 Engineering Specifications

Prototype Dimensions	6” x 7”
Projected weight	~ 600 grams
Screen Refresh Rate	~ 30 HZ
Screen Resolution	**128 x 64 pixels per panel**
Audio Output Power	
Battery life (AA)	2+ days
Charge time	~ 7 hours
Power Consumption	TBD
Controller Feedback	**Haptic technology**

Table 1 Technical Specifications of device

Listed above are specific engineering specifications we plan on meeting for our final design. Due to the limiting factor of time when demonstrating our final product we will not be able to display things like power consumption and battery life. However, three components we will be able to demonstrate are controller feedback, audio, and screen resolution which are bolded above.

2.7 Hardware Block Diagram

Regarding the hardware block diagram, the integral aspect of our design revolves around the microcontroller, which serves as the central hub for all input components within the system, as depicted in the diagram provided. The microcontroller plays a pivotal role in not only orchestrating the connections but also in processing and managing the input signals, primarily stemming from the toggle device. The microcontroller's interaction with the input components is a fundamental element of our project's functionality. It acts as the bridge that facilitates bidirectional communication, receiving input from the connected devices, most notably the toggle. This input is then subjected to the execution of our programmed code, which resides on the microcontroller.
The code embedded in the microcontroller operates as the decision-making brain, dictating the course of action in response to the input it receives. It formulates precise instructions that govern the behavior of both the screen module and the speaker. These instructions dictate not only what actions to perform but also the timing of these actions.

In essence, the microcontroller serves as the conductor of an intricate symphony, orchestrating the harmonious interplay between the user's inputs and the tangible feedback provided through the screen and speaker modules. This symbiotic relationship ensures that the user's interactions are translated into meaningful responses, thereby delivering an immersive and responsive experience in line with our project's objectives.

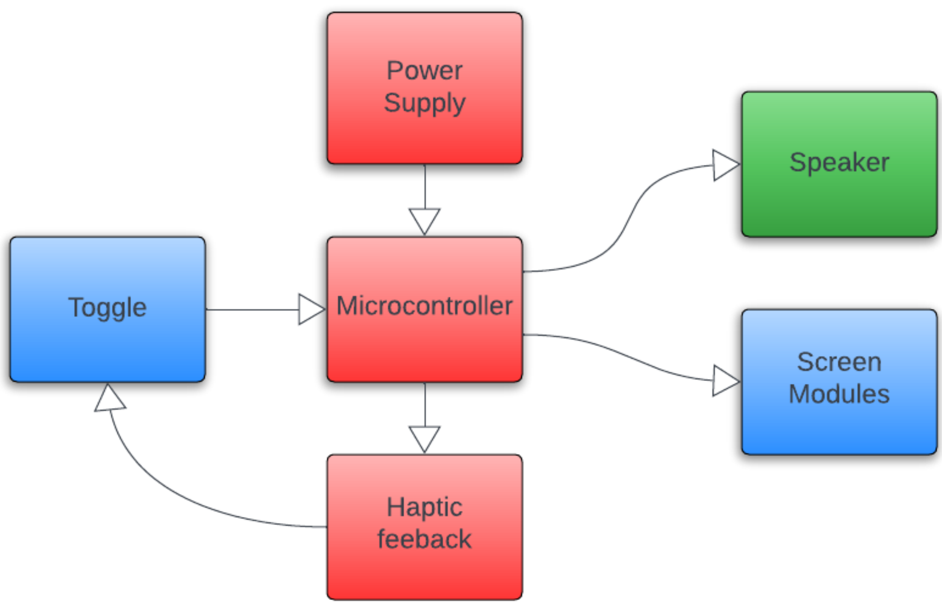

Figure 3 - Hardware block diagram

2.8 Software Block Diagram

Regarding the software block diagram, the team is in the initial stages of establishing the programming structure for the product and defining the required functions. It's envisioned that the software will be designed to operate with different "zones" to track the user's interactions effectively. This zoning approach is aimed at facilitating a checkmark system, which will be utilized to monitor the user's progress and pathing.

The core concept revolves around the user's interaction with a toggle, which needs to be moved in the direction indicated by an arrow. The user is expected to navigate the toggle through a series of zones in a specific order, as indicated by the arrow's direction. The purpose of these zones is to verify if the user successfully passes through them in the prescribed sequence. Once the toggle successfully traverses all three zones in the correct order, a counter will keep track of how many arrows have been cleared in that particular stage. The completion of a stage is determined by comparing the number of cleared arrows with the number of initially illuminated arrows. When these two quantities are equal, the stage is considered complete. This process will repeat for each subsequent level.

If the user fails to complete a level before a timer runs out, the system will initiate a check to determine whether a new high score has been achieved. This is done by comparing the current score with the previously stored high score. If a new high score is attained, it will replace the previous record, and the system will reset to its initial state, enabling a fresh game to begin. If no new high score is achieved, the system will skip the high score update and proceed with a system reset in the same manner as if a new high score had been established. Throughout the user's gameplay, a counter will be employed to keep track of the current level, which will be incremented as the user progresses through the stages.

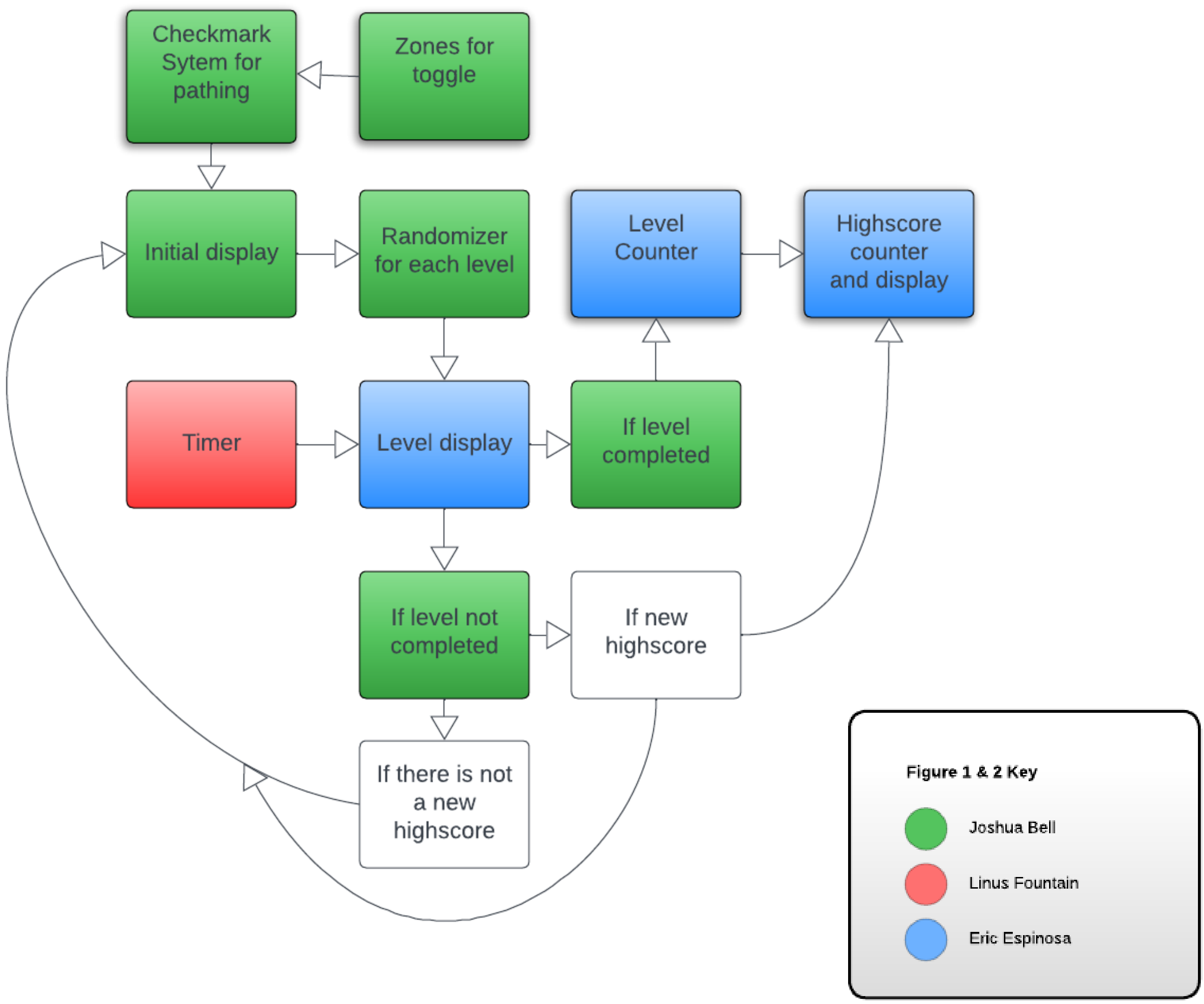

Figure 2 - Break down of all input and output components to be installed

Block Description	Group Member Assigned	Block Status
Power Supply	Linus Fountain	In progress
Haptic feedback	Linus Fountain	In progress
Toggle	Eric Espinosa	In progress
Screen Modules	Eric Espinosa	Completed

Microcontroller	Linus Fountain	Completed
Speaker	Joshua Bell	In progress
Zones for toggle	Joshua Bell	In progress
Checkmark System - pathing	Joshua Bell	In progress
Initial display	Joshua Bell	In progress
Highscore counter and display	Eric Espinosa	In progress
Level Counter	Eric Espinosa	In progress
Level display	Eric Espinosa	In progress
Timer	Linus Fountain	In progress
If level not completed	Joshua Bell	In progress
If level completed	Joshua Bell	In progress
Randomizer for each level	Joshua Bell	In progress

Table 2 - Represents who is assigned to acquire the component or complete the task. As well as the status of the component or task

The representation provided above shows the distribution of responsibilities among our team members, with each individual playing to their respective strengths. Given the varying specializations within the group, those more inclined towards hardware components have naturally chosen tasks of that nature. Conversely, a member with a strong background in software development and programming will be taking the lead in these areas.

It is important to note that while we have tasks assigned to specific team members, our project is anticipated to be a highly collaborative effort. We recognize that the fluid exchange of ideas, insights, and assistance among all group members will be integral to the project's success. This collaborative spirit will ensure a well-rounded approach, allowing us to tap into the collective expertise of our team as we work harmoniously towards our common objectives.

2.9 House of Quality

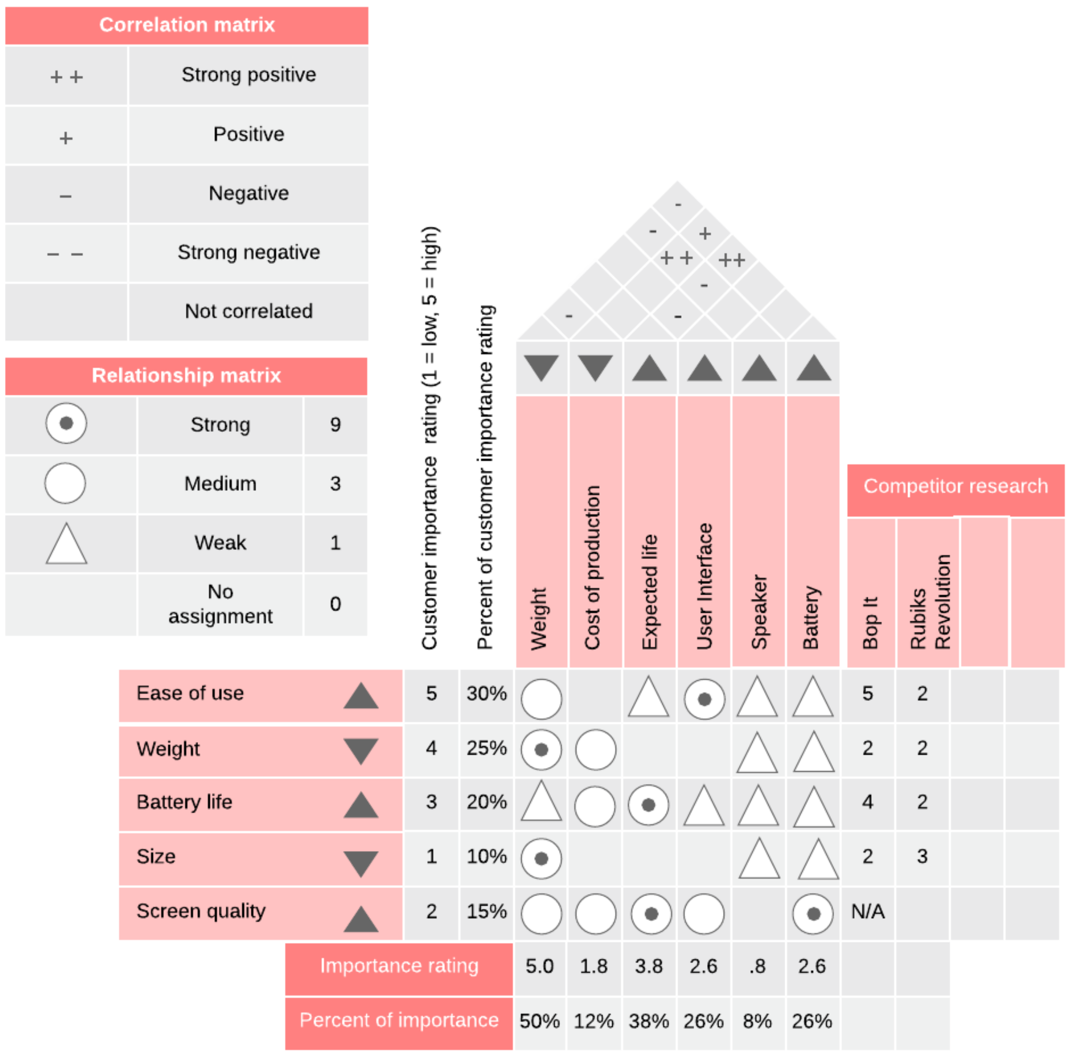

Figure 4 - House of Quality

Above, you can see our House of Quality, which serves as a visual representation of our project's key objectives and priorities. We've placed a strong emphasis on ensuring that our device is not only portable but also user-friendly, catering to a broad audience. This is clearly reflected in our prioritization of "ease of use" as our highest-rated standard, with "weight" closely following as a significant factor. However, it's crucial to emphasize that while we prioritize portability and user-friendliness, we remain committed to maintaining the overall quality of the device. Our goal is to strike a balance that allows us to deliver a high-quality product that is accessible and convenient for all users.

Table 4 - Estimation of milestones until finish

Chapter 3: Research

3.1 Display

In order to bring our project to life, it becomes imperative for us to select an appropriate display that will serve as the canvas for showcasing the main gameplay of our game. At the start of this decision-making process, we find ourselves presented with a plethora of options, each with its own unique advantages and considerations. Therefore, the following section will go into the collective thought process of our team, outlining the extensive research that will ultimately guide us toward making a well-informed choice regarding the type of screen that is most likely to maximize the chances of success for our project. By thoroughly exploring the available display alternatives and carefully weighing the pros and cons associated with each, we aim to ensure that the display we ultimately select aligns perfectly with our project's specific requirements and goals.

When deciding what type of display to have for this project we looked into a few different possibilities for what type of display we will be using. We have had the idea of going extremely basic and having just single led lights under a clear plastic top that would be in the shape of an arrow. This would work well and be extremely cheap however we realized we would have issues with colors bleeding into one another. We also felt that while this would be a cost effective option we thought it would look cheap which we wanted to avoid. We then moved onto the idea of having small individual screens that we could program to have arrows on each screen. This would work by having all of the screens programmed together so that they could display the game properly. However we wanted to have the option to show scrolling text on the screens so this did not prove to be a viable option. Another consideration we had when deciding which display to use was we wanted to have the display be curved. We thought that this would provide a clean look to the user and allow for a unique look of the project to make our product standout. However, we quickly realized that a curved display that would fit the exact dimensions we desired is not readily available so we shifted our focus. Not being able to have the curved screen led us back to having one large screen that would contain all of the display, or having multiple smaller screens connected together. For the smaller screens connected together we had a few different options. The first, cheapest option is an lcd screen Universal-solder Electronics Ltd 1502 2 x 16 Blue-white. It is available for only $3.66 per screen. Its total dimensions are 80.00mm x 36.00mm x 11.00mm, it’s estimated we would need about 4-5 of these screens bringing the total length of the screen to ~ 352mm - 440mm. There is also the 16x2 Blue LCD Module Controller HD44780 that is actually designed for Arduino and raspberry pi. This could be a very appropriate display to have because of that specific connection. This screen is slightly more expensive at $5.99 which could add up if we need to purchase multiple screens. It is also slightly larger being 80mm x 36mm x 15mm. For the concept of having one large screen we have more limited options as there are not an abundance of larger screens that we could find that are available for an Arudino or MSP chip. Our first option is a 7 inch piece found on amazon, that is 192mm x 96mm, vastly larger than the previous options. It is however $33 so it could get pricey if we are running tests and need to order multiple of the screen. Also another factor is display quality, this specific board is a matrix panel so you can see each individual pixel and may not look great especially with moving lights all across it. Since we want to have better display quality then a matrix panel, we looked at a 7 inch TFT Display. This device has a much better display quality at a pixel density at 800 x 480 and is actually slightly cheaper at $29.95. However, this device recommends a companion driver board in the Adafruid RA8875 to be able to connect it to external Microctontrollers. This device is another $39.95, so the total to use this display comes out to about $70.00. Another issue with this display is that the cheaper version that would be ideal is out of stock currently. They do have another display that is exactly the same, however it is touchscreen and therefore more expensive. IT being more expensive and touchscreen which would add another layer of difficulty takes this device out of contention.

	Universal-solder Electronics Ltd 1502 2 x 16 Blue-white	16x2 Blue LCD Module Controller HD44780	7” Matrix Panel	7” TFT Display
Color variety	Blue background, white text	Blue background, white text	24 bit color display	24 bit color display
Price	$3.66 x 5	$6.00 x 5	$35.49	~ $70.00
Dimensions	80.00mm x 36.00mm x 11.00mm	80mm x 36mm x 15mm	7.56 x 3.78 x 0.59 inches	6.5 x 3.9 0.2 inches
Pixel density	N/A	N/A	N/A	800 x 480

Table 5 - LCD Screen Comparison

After further research we have decided that the smaller Universal-solder Electronics Ltd 1502 2 x 16 Blue-white and 16x2 Blue LCD Module Controller HD44780 screens will not be an option as they only have the option of having white text and our design relies heavily on being able to choose multiple colors. If we did decide we wanted to do this design we would have to change the original idea of having colors involved with the system. We could potentially change the idea to having shapes but that is not currently a real consideration.

With more consideration and discussion there was some realization that our main options were the 7” TFT Display and an identical but smaller 4.3” TFT display. These become the center of our research. Both of these components have identical specifications except the fact that one is 2.7 inches larger and as a result has a slightly smaller pixel density at 480 x 272 compared to the larger 800 x 480. We originally were trying to find the largest screen possible that was also compatible with the raspberry pi however we realized shortly after that the smaller screen may be best for a few reasons. The first being that it is cheaper, the 4.3” screen is $29.95 compared to the $34.95 7” screen. We also wanted to consider the size difference in the perspective of a user. The 7” screen would be great because it would allow for large images and high quality, however we decided that we did not need to have such a large screen for such a simple game. This would force us to fill up extra space with clutter and extras that may take away from the games sleek design and may take the users focus away from the actual gameplay.

With all of these considerations we believe that the 4.3" TFT display will allow us to have the best product going forward. It will allow us to have the possibility for every one of our goals that is related to the screen while also being inexpensive, available, and high quality.

3.1.1 Display Driver

One thing that we also may be required to obtain for the screen to work as intended is a recommended driver board. This board known as the RA8875 driver board is recommended by adafruit. The board allows direct connection onto the development board as there is no way to connect our display. Our display will require a 60Hz refresh and 4MHz pixel clock that is not supported by the arduino that we will be trialing. The arduino also does not even consist of enough pins to connect the screen. These conditions lead to adafruit to recommend adding this board to your project if you plan to test with an arduino board to greatly increase quality and simplicity. This board also has a basic graphic library that we can download for it which will help tremendously on all programming fronts. The way that this board works is it allows the screen's ribbon cable to be directly connected to the input port. Then on the RA8875 board there is a driver chip and also 768KB of ram that will allow the board to buffer the display.

Figure 6 - Screen Driver

This product will help dramatically increase the quality of our product and also dramatically increase our quality of life when using the corresponding screen. However we must consider the price point of $34.95 and also where we would put the board inside of our controller. The price point while slightly more than expected is in fact something we will consider reasonable for all of the capabilities it provides. As for the size of the device, there are no specific measurements of the device available in its datasheet or description, however it is shown that it is not much larger than a quarter which should fit nicely into some of the different models of our device. After considering the price of the board and the size that it will take up in our device we have deemed this product to be essential for our project. We believe that this board will allow us to produce the most high quality and professional look for our end device.

3.2 Microcontroller

The microcontroller, often referred to as the MCU, will serve as the central nervous system of our device, orchestrating and overseeing every part of our project. Its primary responsibility lies in executing the code we program onto it and transmitting precise commands to each individual component. It is imperative that we meticulously select a microcontroller capable of shouldering the multitude of tasks we assign to it.

This microcontroller must possess the strength to seamlessly manage a variety of functions. First and foremost, it must support the operation of a screen, ensuring that it runs at a commendable frame rate of no less than 20 frames per second. In addition, it should seamlessly accommodate a small speaker, enabling audio feedback and enhancing user interaction. The inclusion of an analog or digital control toggle is also a prerequisite, offering intuitive and versatile control options. Moreover, our microcontroller must be equipped with the necessary memory resources to store essential data, such as high scores for our game. This ensures that players' achievements are preserved and can be accessed seamlessly.

In essence, selecting a microcontroller that can proficiently handle all these multifaceted processes is crucial to the success of our entire system. Without this core component operating seamlessly, our vision cannot be realized. The subsequent section delves deep into the intricate aspects and responsibilities this microcontroller must shoulder, underlining its paramount role in the realization of our project.

3.2.1 Microcontroller Options

This day and age, there are a magnitude of different microcontrollers to pick from, all varying in components, functions, memory, and speed depending on what you are looking for in the microcontroller and project. Researching the different microcontrollers, we have stumbled upon some that we thought would work well for our project and idea in mind. The ones we have considered to pick from are:

- Raspberry Pi Zero W - Broadcom BCM2835 SoC
- ESP32 - Espressif ESP32 Microcontroller
- MSP430FR6989 Microcontroller
- Arduino Board - ATmega328P Microcontroller

3.2.1.1 Raspberry Pi Broadcom BCM2835 SoC

The Raspberry Pi Broadcom BCM2835 SoC is a remarkably compact and budget-friendly microcontroller that has been at the heart of a wide range of innovative projects. Its adaptability knows no bounds, allowing it to be transformed into diverse creations, including a security camera, portable media center, Wi-Fi hotspot, or even an internet radio station. The versatility of this microcontroller makes it a top choice for enthusiasts and tinkerers seeking to bring their imaginative concepts to life. It was among the very first microcontrollers that came to mind, thanks to its extensive capabilities and the creative possibilities it offers. One specific project in mind that made this one of our options is that Raspberry Pi's can be used to make a retro gaming console by using software like RetroPie. This would allow the user to play games from consoles such as the Nintendo Entertainment System (NES), the Super Nintendo Entertainment System (SNES), and Sega Genesis, all of which are more graphic and processor intensive than our project. So, we know this microcontroller would be able to handle our design.

The model we considered using was the Raspberry Pi Zero W Broadcom BCM2835 SoC. This microcontroller model is powered by a 1 GHz single-core ARM-S processor, which we believe is more than enough for our project. The controller has 512MB of RAM that is shared with the on-chip GPU, making it suitable for running basic applications and projects. This board features a mini-HDMI port and micro-USB, giving us more options for our components and back up selections. It also features onboard WI-FI and Bluetooth 4.0, which won't be needed for the core of our project, but this allows us to expand and add more features if needed. The Broadcom BCM2835 SoC can run various operating systems and supports a wide range of languages including Python, C/C++, and more. This is great because our group is more familiar and comfortable with C/C++, so we would not have to spend extra time learning a new language. It is also known for its low power consumption, making it suitable for our handheld, battery powered project. The Broadcom BCM2835 SoC is one of the most cost-effective Raspberry Pi controller models making this one of our top contenders for our project, especially since one of us owns a Raspberry Pi Zero W already, saving on costs.

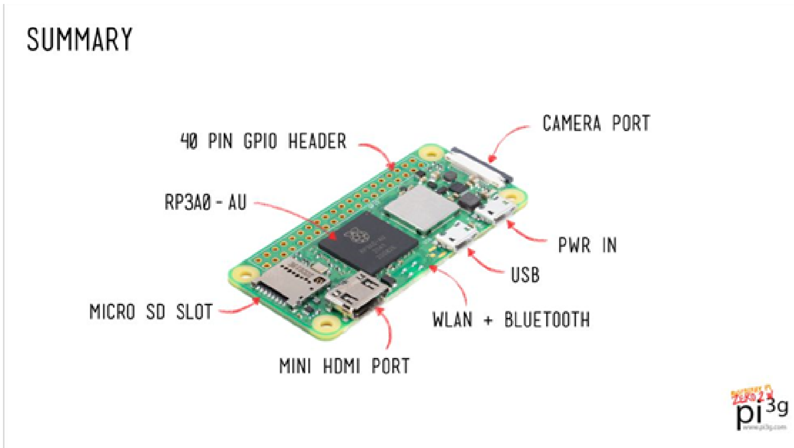

Figure 8 - Raspberry Pi

3.2.1.2 ESP32-WROOM-32 Espressif ESP32 Microcontroller

The ESP32 is indeed an intriguing series of microcontrollers known for their cost-effectiveness and power efficiency. These compact boards come equipped with integrated Wi-Fi and Bluetooth features, making them quite versatile for a broad spectrum of projects. They can be used for creating simple games, such as 2D and retro style games, making them suitable for our project design. Among the array of applications that the ESP32 is particularly suited for, Internet of Things (IoT) projects stand out. Smart thermostats, door locks, and weather stations represent just a few examples of IoT applications that can be enhanced with the ESP32, offering remote monitoring capabilities. Similarly, home automation enthusiasts appreciate the ESP32 for its ability to control various appliances and security cameras with ease. The reason we are considering this one as an option is because the ESP32 has been used to play games such as Pac-Man, Space Invaders, and platformers.

This microcontroller features a dual-core LX6 microprocessor, each core can be clocked to up to 240 MHz, which allows for multitasking and handling real-time tasks efficiently. The dual-cores would be great for game processing. The Espressif ESP32 microcontroller includes 520KB of SRAM, which is divided into RAM, data RAM, and other memory. It also has 448KB of ROM, which is used for system functions and initializing code. The Espressif ESP32 microcontroller does not include a dedicated hardware for graphics, but is able to use libraries and functions to draw images on the screen. The microcontroller can be programmed with a variety of tools and languages, including Arduino IDE(C/C++), MicroPython, and PlatformIO. Of these we are familiar with the Arduino IDE, since it is a C/C++ based IDE. The costs of the board range from $5-$10, making it a very affordable option.

Considering all these specifications, we decided that the Espressif ESP32 would be a good option as a microcontroller, but we will choose not to proceed with it considering we have other microcontrollers already in our possession, saving us time and money to use the options available to us already. This is a good microcontroller to keep in mind, and we can order one as well as a backup.

3.2.1.3 MSP EXP430FR6989 Microcontroller

The MSPEXP430 proved to be a practical choice for our project, aligning seamlessly with the familiarity we all share from our past engineering courses. Opting for this board would benefit us by not only reducing our project costs, but also get around the delays associated with purchasing and shipment of the board. After researching this microcontroller, it was revealed that it does not excel in handling graphically intensive games. Nonetheless, it can be used to create simple or lightweight games, such as Pong, Snake, Simon Says, and some simulation-based games. We believed it held the potential to fulfill our project requirements.

This microcontroller features a 16-bit RISC CPU, which operates at a maximum clock frequency of 8 MHz. This is not designed for high performance computing but is practical for many low powered and embedded applications. The MSP comes with 128KB of Flash memory for program storage and 2KB of RAM for data storage, this is used for storing code and data. This board offers a wide range of integrated peripherals, none of which would be useful for our project. The microcontroller uses Code Composer Studio, which is C/C++, which we are familiar with. Since we each own our own MSP430 board, we would save on costs for purchasing and shipping.

In summary, this microcontroller does not meet our expectations for our needs. We had initially believed that it would suffice, and fulfill our needs, but the limitations of a low processing power and memory give rise to concerns regarding its ability to meet our requirements. It could potentially address our needs, but just barely. So the prospect of investing time and effort into utilizing this board where success is not certain, would not be a worthwhile choice when there are other options available, where success is more attainable than using the MSP430.

3.2.1.4 Arduino Uno ATmega328P

Arduino's represent an exceptional and open-source microcontroller platform, offering a remarkable degree of versatility for creating electronic projects. These devices are designed for a wide range of enthusiasts who want to build and prototype various systems. What truly stands out about this diverse board and chip series is the expansive array of options it gives us, allowing for a spectrum of creative possibilities to be explored. Among these choices, our preference leans towards the highly regarded Arduino Uno, which came recommended by many of our classmates and previous Senior Design class takers. However, should the need for more capabilities or expanded resources arise, we have other microcontrollers and options available to us.

The Arduino Uno is powered by the ATmega328P microcontroller that operates at 16 MHz, which is sufficient for our tasks. It has 32 KB of Flash memory for storing program code, 2KB SRAM for data storage and 1KB of EEPROM for non-volatile data storage, this gives the Uno a versatile memory hierarchy. The Uno lacks a dedicated GPU, but it can interface with various display modules and sensors for graphic display. The ATmega328P has various peripherals and

I/O pins we can use, making this a great option for us to connect our toggles, speakers, screen, and haptic feedback. The ATmega328P uses the Arduino IDE for programming, the IDE is user friendly and supports C/C++. And luckily, one of us has an Uno in our possession, saving us costs and time. After comparing and researching boards, the Arduino Uno is a cost-effective microcontroller that meets most of our criteria and needs

3.2.1.5 Microcontroller Comparison Table

	MSP430FR6989	**Arduino Uno**	**Raspberry Pi Zero W**	**ESP32-WROOM-32**
Microcontroller	MSP430FR6989	ATMega328P	Broadcom BCM2835 SoC	Espressif ESP32
CPU	8-Bit	8-Bit	16-Bit	32-Bit
Core Count	Single	Single	Single	Dual-Core
Clock Speed	16 MHz	16 MHz	700 MHz	240 MHz
Integrated GPU	No	No	Yes	No
RAM/Memory	2 KB Ram/128 KB Flash	2 KB SRAM/32 KB Flash/1 KB EEPROM	512 MB RAM	540 KB SRAM/448 KB ROM
Environment	Code Composer Studio	Arduino IDE	Raspberry Pi OS	Arduino IDE, PlatformIO, MicroPython, and the Espressif IoT
Coding Language	C/C++	C/C++	Python, C/C++, Java, Ruby	C/C++, MicroPython, JavaScript
I/O Pins	83 I/O pins	23 I/O pins	40 I/O pins	38 I/O pins
Voltage Range	1.8V to 3.6V	5V to 12V	3.3V	3.3V
OperatingTemp	-40°C to +85°C	-40°C to +85°C	0°C to 70°C	-40°C to +85°C
Price	$0	$0	$0	$5-$10

3.2.2 Microcontroller Considerations

After a thorough examination of the available boards, weighing the advantages and disadvantages of each, while considering our project demands and functionality, we have decided between the Raspberry Pi Zero W and the Arduino Uno. Both of these can meet our criteria with relative ease, but a more in-depth analysis is needed to see which MCU will align with our requirements.

3.2.2.1 Display compatibility

The display quality will be largely dependent on which microcontroller we have running the device. A weaker microcontroller, for example, may not have the ability to run a display at the frames per second that we desire. Because of this we need to make sure that we choose a microcontroller with enough ability to run such a screen. By doing so, we can eliminate any devices that fall short of the requirements with such limitations.

However, this does not mean that we must choose a microcontroller at the extreme end of the spectrum that can run a display upwards of 100 frames per second. Such a choice would be excessive and most likely would place undue stress on the power supply and be more costly in the long run. We want to find a middle ground between the two in order to stay under a reasonable budget but also still provide the highest quality device we can.

The ATmega328P is commonly used in various projects with displays. It is capable of handling basic displays, but has certain limitations when it comes to handling graphics and display related tasks. It may struggle to achieve a smooth frame rate for complex games, but our project idea game design is not complex graphically. The ATmega328P is not suitable for high frame rates, and is more commonly used with low resolution displays. It lacks processing power and memory for fluid graphics, as well as built in video interfaces and may require additional components and drivers.

The Broadcom BCM2835 SoC offers significantly more computational power and graphics capabilities than the ATmega328P. It is better suited for running a simple handheld video game with a frame rate of 20 frames per second on an LCD screen. It offers a good balance of computational power and graphical capabilities. It can handle a variety of display resolutions, but not high-resolution displays commonly found in smartphones or tablets. It also supports HDMI and composite video outputs.

3.2.2.2 Control toggle compatibility

Another critical component that we must ensure can be seamlessly controlled by our chosen microcontroller is the control toggle, a pivotal aspect of our project. Within the section discussing control toggle options, we explore two main types that are under consideration. The type of control toggle we ultimately select will significantly influence our decision regarding the microcontroller best suited for our project.

This seems to be a very key function in most microcontrollers, however if a microcontroller for whatever reason does not have this capability, then we will immediately exclude it out of consideration for this project. The control toggle will be an essential component of our device operation and therefore if a chip is unable to handle it, we cannot use it. Fortunately, as previously stated, most microcontrollers that we research have some sort of control toggle used with it, making this a highly unlikely hurdle in our selection process with microcontrollers.

Both the Broadcom BCM2835 SoC and ATmega328P can be programmed to control and interact with various external components, including toggle switches. So we can connect a physical toggle switch to them and program it to respond to certain inputs.

3.2.3 Microcontroller Familiarity

A key factor to consider that guides us in choosing between microcontrollers is our familiarity with these chips. As a group our prior exposure to microcontrollers in the past has been very limited, so it is important to us to find a chip that we have at the very least heard of before. We do not want to go into this project with a complete lack of knowledge about every single aspect of a microcontroller so it is imperative we decide on a microcontroller that we either have had previous experience with or a popular chip that we can gather resources on if possible.

For example, we all have experience with the microcontroller from the MSP430FR6989 Launchpad board. Since we all have previous experience with this chip, we have a basic understanding of what this microcontroller is capable of, how to use the software that goes along with it and the fundamental principles on how to program the chip. Our ability to use these skills already will at the bare minimum save crucial time that we can use on other aspects of the project. Of course, we will still have to learn many new techniques and skills but having a base knowledge, or an easy way to obtain a base knowledge, of the microcontroller will play a heavy role in our final decision.

Due to the specifications and limitations of the MSP430FR6989, we have decided to pick between the Broadcom BCM2835 SoC and ATmega328P microcontrollers for our project. We are not too familiar with them, but feel comfortable and confident enough to use them. They both can run Arduino IDE, which is a C/C++ based language that we are most familiar with. And they both have big communities behind them that we would be able to seek guidance in case things do not go according to plan.

3.2.4 Microcontroller Memory Capability

Something that we will have to keep in mind when picking a microcontroller is its storage capabilities. Part of what makes most games fun is the competitive aspect of them and in order to make our game give players a competitive drive, we want to have high score data stored within the device in order to provide a goal for the users when they use our device.

We took inspiration from older arcade games and pinball machines as a palace to look at what type of memory we may need. We found that Electrically Erasable Programmable Read-Only Memory (EEPROMs) is a popular way to achieve what we are looking for as well as Non-Volatile Random Access Memory (NVRAM). EEPROM is a type of non-volatile memory that can be electronically erased and reprogrammed. As previously stated, this is a popular type of memory that is found within arcade games as it is easy to retrieve data from and does not need a battery backup to maintain data during power-off situations. As for NVRAM, it is similar to regular RAM, however, it combines features of RAM and non-volatile storage allowing data to be written to it and read like normal RAM. It retains data when power is removed which is essential to our project because we do not want the high score data to be lost when the device is powered off. There is also a third type of plausible memory we could use as well. That memory would be flash memory. Flash memory is another non-volatile storage that retains data even when powered off which is essential for this part of the project as previously stated.

Fortunately, many microcontrollers designed to meet these requirements come equipped with at least one of these options integrated onto the chip, making the selection of a microcontroller relatively straightforward. This will allow us to freely select a chip and double check at the end that our chip has one of these options as it more than likely will.

The ATmega328P has only 2 KB of SRAM, making it a challenge to store graphics, game logic, and other game data, but this should not be an issue with our game, since it will not be graphics intensive. It also comes with 128 KB of flash RAM that is a necessary component for our game design, since it will be required to store data once the game is powered off.

The Broadcom BCM2835 SoC comes with 256 MB of RAM making this a great option for any form of game below simple 2D games, and emulators for older games. It may be limiting for more complex 3D games, and not suitable at all for newer generations of games, but playing these graphic intensive games is not necessary for our project, as our gameplay will be at the level of a 2D game or less. The downside to this microcontroller is that it does not include flash memory, which is necessary for our design. In order to overcome this obstacle, we will have to purchase and incorporate a microSD or external storage, as these have flash RAM included.

3.2.5 Final Decision

Going from a few microcontrollers from the MSP430FR6989, ATmega328P, Broadcom BCM2835 SoC, and the ESP32-WROOM-32, we narrowed it down to the ATmega328P and Broadcom BCM2835 SoC. After thoroughly researching and comparing between the two, we have decided that the Raspberry Pi's Broadcom BCM2835 SoC would most likely be a better fit for our project. After researching the Raspberry Pi's Broadcom BCM2835 SoC is the better choice, with better graphical performance, better processing speed, better components, and for the same cost. Yet, we are still adamant on at least testing the Arduino Uno's ATmega328P as a personal project. This was our original pick for microcontroller and had our plans originally planned for it, so we wanted to see how it would test the Broadcom BCM2835. They both run Arduino IDE, making the coding nothing more than a simple transfer with a few function changes here and there.

Final Microcontroller comparison		
	Broadcom BCM2835 SoC	ATmega328P
Price	★★★★★	★★★★★
Display Compatibility	★★★★☆	★★☆☆☆
Memory Availabilty	★★★★☆	★★★★★
Toggle Compatibility	★★★★★	★★★★★
Group Familitarity	★★☆☆☆	★★☆☆☆

Table 6

3.3 Control Toggle

In the realm of control toggles, there exist two primary alternatives deserving our consideration: analog input and digital input. The first choice, analog input, extends a sophisticated avenue that enables users to execute movements with utmost precision, granting them the capability to manipulate their position either gradually or rapidly. This methodology takes into account not only the direction indicated by the user but also the extent to which they point in that direction. In contrast, the digital input method can be likened to a set of discrete buttons concealed beneath the control stick, which are activated as the stick is manipulated. This approach results in a binary outcome, either pressed or not, thus offering a diminished degree of finesse in user movement control.

Analog Toggle	Digital Toggle
High power consumption	Low power consumption
Lower Speed of operation	Higher speed of operation
More Expensive	Less expensive

Table 7

Upon careful analysis, it becomes evident that the latter option aligns more closely with our specific requirements. In our case, the emphasis lies not in attaining pinpoint precision but rather in discerning the user's intent to shift in the left or right direction. Consequently, the digital input mechanism proves to be the ideal fit for our objectives. By focusing solely on detecting leftward or rightward movements without the need to factor in distances, we simplify the programming process considerably. This streamlined approach aligns perfectly with our project's goals, fostering efficiency and ease of implementation.

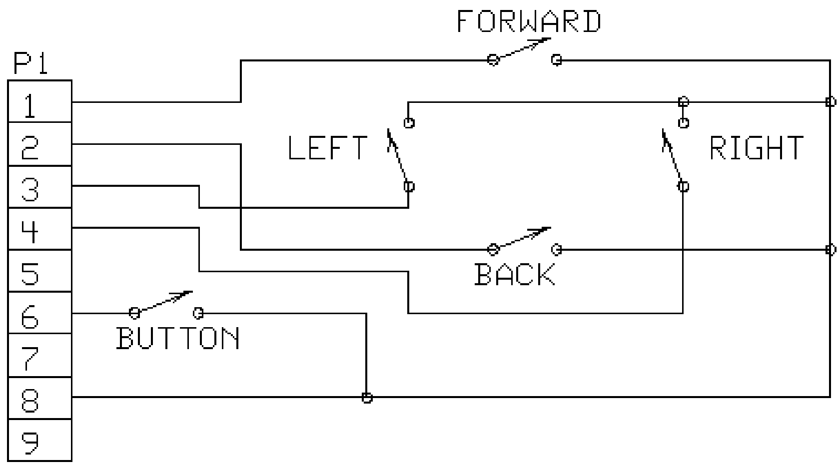

Figure 10 - Toggle Schematic

Above is a schematic of a digital joystick that clearly shows the switches that it uses for left, right, up and down. The schematic shows that when the stick is pushed into one of those positions, the connection is completed allowing for the electrical signal to flow and thus completing the circuit for that specific direction.

3.4 Sound Device

The sound device's only action will be to play a timer that plays a single beeping sound that will continuously get faster as each level progresses. This is a very simple task so we will be looking for a very simple device to achieve this. We have decided that this is all the sound device will do in order to save on space, battery life and power usage. When considering which speaker to use in our product we are considering all of the previously mentioned things. The sound device is an integral part to the game because it allows the user to have an idea of when the level will be over however we want it to only be a small component. We need to look at devices that will not take up a lot of room in our device and will also not drastically affect the power consumption and the battery life. When taking a deeper look and comparing individual speakers we mainly considered the Soberton Inc. SP - 3605, Adafruit industries LLC 4227, and Soberton Inc. SP-2804L. The first thing we looked at when trying to decide between all the possibilities online was the price point. The three of these are all less than $3 so the price is close to negligible. This is ideal because we can order a few of them and test them against each other to see which will work best. Next we considered compatibility, all of them have the same type of connection, that being two wired ends. We believe this to be ideal because you can do different things on the end of those to make it connect to the final design. Next we considered the power consumption. For the SP - 3605 we have 1W of power consumption, the Adafruit industries LLC 4227 we have 1W of power consumption and the SP-2804L we have 1.5W of power consumption. After the consideration of power consumption the dimensions would be the next specification to consider. The SP-3605 is 36 x 5.0mm, the Adafruit industries LLC 4227 is 30.0 x 20.00mm and the SP2-804L is 28.0 x 4.4mm, the overall dimensions by all parts should allow any of them to fit nicely into the base of our controller with the Adafruit industries LLC 4227 being the largest.

	Soberton Inc. SP - 3605	**Adafruit industries LLC 4227**	**Soberton Inc. SP-2804L**
Price	< $3	< $3	< $3
Power Consumption	1W	1W	1.5W
Size	36 x 5.0mm	30.0 x 20mm	28.0 x 4.4mm

Table 8

3.5 Haptic Feedback

Due to the nature of haptic feedback we are going to have to order some sample modules to test for what kind of feedback we think will suit our device the best. Luckily they are very inexpensive so it's easy to order a few and test them quickly to see if they are what we want. Design wise we want something that's firm and "clicky" so when the user is moving between the sections of the game it feels almost as if they are spinning a physical dial. But nothing so strong as to be like a rumbler that goes into gaming console controllers.

For haptic feedback motors there are two popular designs: one for electric toothbrushes (left) and one for mobile phones (right). The electric toothbrush design is (allegedly) a lot stronger than the mobile phone design. While the haptic feedback of a mobile phone is the ideal, our device will more than likely be a good bit larger than a phone, and thus might require a stronger module for the user to feel the same result.

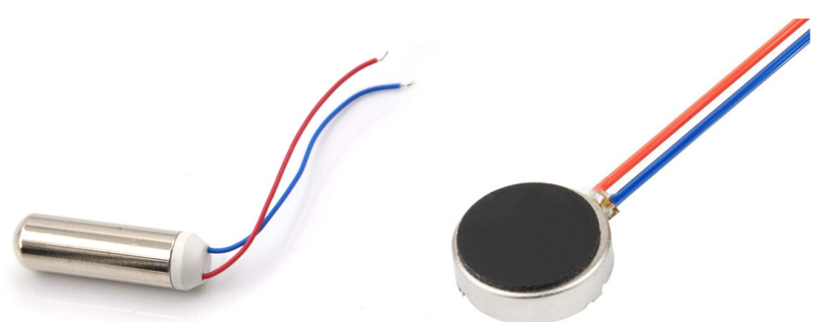

Figure 11 - Haptic Feedback Modules

3.6 Battery

For the battery we will have to choose based on the final decisions of our other components, but we expect to only use something like a few double A batteries or maybe a 9V battery. We don't expect any of our components to use a lot of power as none of them are large motors or anything like that.

3.7 Software and Programming Language Selection

Selecting the appropriate software and Integrated Development Environment (IDE) is of paramount importance in engineering projects. The choice of software and IDE significantly influences project efficiency, accuracy, and the ease of collaboration among team members. The right tools not only streamline the design, simulation, and analysis processes but also facilitate effective debugging and testing. Moreover, they ensure compatibility with hardware components, enabling seamless integration and efficient utilization of resources. Making the correct software and IDE choices optimizes project development, reduces errors, enhances productivity, and ultimately contributes to the successful and timely completion of engineering endeavors.

When deciding between which software and even programming languages to use our team really wanted to use something that we were familiar with. We decided that the difficulty of learning an entirely new programming language was unnecessary given the options that we had available. When deciding which programming language to use it is heavily dependent on what MCU we will use. These two decisions went hand in hand as each MCU uses their own specific programming language. The Arduino Uno uses a variation of C# as well as the Raspberry Pi since they both can share the same Arduino IDE.

After the selection of our MCU and programming language we must look at the softwares to use for our MCU. Arudino does in fact have its own IDE that is recommended, however we wanted to consider all options. Some other popular options are Microchip studio, Visual Code Studio (with the PlatformIO extension) and avrdude. Microchip studio is a great IDE however it has a very steep learning curve that we may not want to take on or have the time to do. Visual code studio is a very plausible option as we are all familiar with it. Visual Code studio also known as VS code is a very popular IDE used by many software engineers. It will definitely be in our final consideration. As for avrdude none of us have ever heard about it before. After further research we have found that avrdude is a software designed for the retrieval and transfer of data in the on-chip memories of Microchip's AVR microcontrollers. This versatile tool can be used to program the Flash and EEPROM memories, and when supported by the programming protocol, it can configure fuse and lock bits. Furthermore, AVRDUDE offers a direct instruction model that permits users to issue custom programming commands to AVR chips, regardless of whether AVRDUDE natively supports a specific feature of a particular chip.

Following an extensive deliberation of our available choices, our current inclination leads us towards embracing the recommended Arduino IDE as the primary programming environment for our project. This decision is based on the anticipation of complete support for our specific microcontroller chip, coupled with the potential wealth of resources and libraries it may provide to help our development. However, it is crucial to acknowledge that, at the time of drafting this document, we have not yet acquired the microcontroller hardware. This circumstance leaves us unable to be entirely sure, which programming platform will best align with our preferences and requirements. In light of this inherent uncertainty, we have maintained an open stance towards Visual Code Studio (VSCode), recognizing it as a viable alternative in our contingency plan. The reason for this consideration comes from our team's collective familiarity and comfort in using VSCode. Should unforeseen challenges or complications arise with the Arduino IDE, we can seamlessly transition to VSCode as a backup option, ensuring an uninterrupted and efficient project development process. This approach affords us the flexibility and adaptability necessary to ensure the successful execution of our project, irrespective of the chosen programming environment.

Final IDE comparison		
	Arduino	Visual Studio Code
Familiarity	★☆☆☆☆	★★★★☆
Library support	★★★★★	★★★☆☆
MCU/dev board support	★★★★★	★★★☆☆
User experience	★★★☆☆	★★★★★

Table 9

3.8 Project Constraints

Building and designing this device comes with certain constraints and limitations, some of those are as follows. One of the constraints is resource limitation. Due to ongoing events, such as the chip shortage, we might not be able to find certain needed parts. Therefore, we have to use what's available to us, in the market, at the moment. Another constraint is our supplies and materials not arriving on time, due to shipping times, Florida storms, or other travel related issues. In order to offset this constraint, we will order the parts needed as soon as possible, because the earlier the better. Some other issues might occur, such as receiving defective parts, or parts malfunctioning during the project. Our solution to this is to order double parts on components we think might have some issues. Due to the size of the design, there might be some size and weight constraints that can affect our project. We might have to increase the size, decrease the size, or adjust certain areas in order to fit the parts. There can be battery constraints as well, relating to size, we have to adjust or make space for the battery source, and ensure our device has a lasting battery life. We expand upon this in much further detail in the design constraints section of this paper.

In Figures 1 and 2, the block diagrams illustrate how the project's hardware and software components will be organized. The hardware components necessary for the project are to be acquired gradually over the semester. As of the time of writing this paper, the team is in the process of determining the specific components required to minimize potential issues in the future.

Chapter 4: Related Standards and Potential Design Constraints

Each engineering endeavor inherently comes with constraints that define its boundaries. These limitations, far from being obstacles, often serve as guidelines, streamlining efforts and enhancing interoperability among autonomously developed systems. This next segment aims to uncover the external constraints and standards influencing this project, while also delving into how these revelations will shape and support its objectives.

4.1 Related Standards

In the following sections you will find public standards that will have a direct impact on the design of this project. There are many advantages historically to following standards however the main benefit we will receive is that it will allow us to avoid redundant work. They will also allow us to increase the project's overall functionality by ensuring that others can easily utilize and comprehend it. Some of the more emphasized standards that are used in this project are USB 2.0, IEC 61508:2010, and IEC 60086-1:2021.

4.1.1 USB 2.0

The Universal Serial Bus also known as USB is an integral part in almost any electrical or computer device. The USB is a worldwide port that is used on many different devices for numerous reasons. There have been different iterations and advancements made to the USB port. The USB 2.0 is the successor of the USB 1.0 and 1.1. The original USB was introduced in 1996, being developed by multiple companies including IBM, Intel Corporation, and microsoft. The invention of the USB port allowed for a standardized way to share data between devices, allowing for huge advancements in the computing world. USB 2.0 is fully backwards compatible with the USB 1.0 and 1.1 however it must follow the speeds of those ports. USB 2.0 was developed to allow for higher transfer rates, when it was release in April of 2000 it debuted a data transfer rate of 480Mbps, however with certain bus limitations it could only reach data transfer speeds of 280 Mbps at the time this was still an increase from the original USB which had data transfer speeds of 12Mbps. Alongside the deployment of much faster transfer speeds the USB 2.0 allowed for plug and play ability for storage devices and multimedia as well as also having support for power sources using USB connectors upwards of 5 V and 500 mA.

When a device is plugged into a USB 2.0 port, it establishes a bidirectional communication link with the host, enabling data exchange and, in many cases, a power supply to the connected device. USB 2.0 uses a shared bus architecture, allowing multiple devices to be connected through hubs while maintaining compatibility with various peripherals. This standardized connectivity has made USB 2.0 an essential and convenient technology for a wide range of applications, including external hard drives, keyboards, mice, and more.

One of the main uses of the USB port is to be able to communicate with another device in order to exchange data. This was the original reason that the USB was developed. This allows for all different types of data to be sent between the two devices that are connected together by the USB. This is an incredibly efficient and convenient way to transfer data between two devices.

Our project involves the utilization of USB 2.0 technology to transfer our game software onto the Raspberry pi board. It is imperative that the Raspberry Pi board has a reliable method for receiving the software, as this is essential for the successful execution of our game. To achieve this, we have opted to use the USB 2.0 port for loading our software onto the board. The reason we chose to use the USB 2.0 port specifically was because access to significantly faster data transfers, thus optimizing efficiency and saving valuable time in the overall development process.

4.1.2 IEC 61508:2010

IEC 61508 stands as the cornerstone in the realm of functional safety, holding extreme importance across a wide variety of engineering industries, and it is equally crucial to the success of our project. Its origin dates back to the early 1990s, driven by the urgent necessity for a comprehensive and universally applicable standard that could effectively address the challenges associated with ensuring the functional safety of electrical systems. Since its inception, IEC 61508 has undergone several iterations and revisions, displaying its remarkable resilience and adaptability in keeping pace with the rapidly evolving landscape of the electrical world.
This standard's evolution over the years reflects its unwavering commitment to staying relevant and effective in an ever-expanding field of electrical technologies. IEC 61508 remains a cornerstone in the pursuit of functional safety, not only serving as a foundational reference for numerous engineering sectors but also offering invaluable guidance and structure to our project, ensuring that safety and reliability are at the forefront of our thought process.

This standard plays a crucial role in ensuring the safety and reliability of systems such as complex electrical systems, programmable electronic components and much more. These systems are abundant in our world. They are found in a wide variety of things such as a small game controller all the way up to aerospace vehicles and many more devices. IEC 61508 aims to keep risks low with these devices by establishing guidelines for risk assessment, safety integrity levels or SILs and the whole lifecycle of safety-related systems all the way from the design to the operation and maintenance. This standard is relevant to our project because it will help guide us to create a product that is safe for public use.

4.1.3 IEC 60086-1:2021

The International Electrotechnical Commission (IEC) serves as a preeminent global standards organization, crafting and updating international standards for electrical, electronic, and related technologies. Among these standards, IEC 60086-1:2021 stands out, particularly in the realm of batteries. This comprehensive standard outlines critical specifications, encompassing performance, dimensions, and labeling requirements. Given our project's reliance on battery power, strict adherence to IEC 60086-1:2021 is essential, signifying our commitment to aligning with international industry best practices and ensuring the safety and reliability of our solution.

Complying with IEC 60086-1:2021 extends beyond mere formality; it underscores our dedication to delivering a safe, reliable, and technologically robust project. Batteries are pervasive in the world of electronics, making adherence to this standard not just a technical necessity but also a means of promoting compatibility and safety. This commitment assures that our project meets international norms and standards, fostering quality, safety, and technological excellence while facilitating seamless integration within the global landscape of electrical and electronic technologies.

The 60086-1:2021 standard is intended to standardize batteries. This is with respect to dimensions, nomenclature, terminal configurations, markings, test methods, typical performance, safety and environmental aspects. Some of these standardizations heavily relate to our project. Stated later on in this paper, specifically in the safety and environmental constraints we will talk about just how important these two aspects are for our project.

4.1.4 IEEE Standard 754

The Institute of Electrical and Electronics Engineers standard 754 is a standard regarding floating point numbers. This standard was established in 1985 as a way to address problems in how floating point numbers were implemented. Before this standard, many floating point number implementations were difficult to read and were not very reliable. This reduced their portability from computer to computer. With this standard being implemented it helped the standard of floating point numbers become consistent throughout engineering, thus increasing portability and efficiency across all devices.

The standard says that floating point numbers should have 3 components. It should have The Sign of Mantissa, The Biased Exponent, and The Normalized Mantissa. The sign of Mantissa is where a 0 represents a positive number and a 1 represents a negative number, the Biased Exponent is a field that can represent either and The Normalized Mantissa is a part of the number that will hold the significant digits in exponent form.

We will be following this standard throughout our project in our programming section. This will be necessary to help keep our program portable in case of needing to switch to another device. This will also help us be standardized, meaning that if the project were to hypothetically get commercialized and other engineers that were not this group were to work on it, then they would also understand our implementation of floating point numbers.

4.2 Design Constraints

Time constraints for this project will be set upon us by the professors and instructors above us. They have laid out different check-ins and milestones that we must complete in order to achieve a passing grade for this class. Since a passing grade is required for all members to graduate it is imperative that we all contribute equally in order to complete these milestones. These milestones are designed to keep us on track so that we can have a completed design by the end of the current fall 2023 semester and a fully completed project by the end of the spring 2024 semester. If we are unable to meet these deadlines we will collectively not pass the class and therefore not graduate, so it is imperative that we budget our time and resources extremely efficiently.

A significant time constraint that requires our careful consideration is related to the design and production of the printed circuit board (PCB). It is imperative to keep in mind that not only will crafting the PCB design be a time-consuming process as we will likely have to redesign the PCB a few times until we are happy with a final design, but there will also be a considerable duration involved in its actual printing and subsequent delivery to our us. We must also consider the chance that we obtain a PCB that we are not happy with and have to completely redesign and ship a new one which will slow down development but we should plan accordingly and incorporate enough time for this to happen. Because of these constraints, we must factor in these time requirements when planning our project's timeline and ensure that we allocate ample time for both the design phase and the production logistics of the PCB.

Going along with the PCB being a significant time constraint we must also consider is the wait for parts. We of course will be ordering almost all of our parts online and so we must wait for them to arrive. This in itself is a time constraint as we must wait for the parts and then we also must test and put together the parts. We will need to account for the possibility that the parts may not work as we need them to so we will have to order more and wait again. We must make sure that we order our parts with plenty of time to test them and make sure that they work. We must also consider the possibility of parts being out of stock, this is not quite as much of an issue as we will be able to research other similar parts but we must still consider this.

While there are many time constraints related to research and design for this project we must also consider time constraints related to the actual operation of the device itself. For starters, when the user is going from one level to another we must consider a time between the levels so the user has enough time to prepare themselves while also making sure the time between them is not too long that they may lose interest. Also going from each level to the next we must consider how the timer will speed up as the game goes on. We may do a linear progression for slightly easier gameplay or an exponential progression for harder gameplay. Finally our device will potentially have haptic feedback within the controllers. A time constraint for that will be we will need to find the perfect timing for the controller to rumble after an input is pressed by the user. We again do not want the delay to be excessive however we do not want it to be abrupt, our group must find a middle ground for this as well. These can also be considered design time constraints as well because they will take time to test and find the perfect setting for. With these time constraints and more we must be able to be self aware and honest with how long these tasks will take us and start earlier enough to account for any setbacks.

4.2.2 Economic Constraints

The main economic constraints that we will face are mainly due to purchasing the components for the device. However, there will be other purchases we must consider and since our project will only be funded by the members of the group we will have significantly more constraints then if there were outside funding for this project. It's important for us to select components that are not unreasonably expensive for us as we do not have a sponsor while also remembering that this is a very important project and in order to be successful we will have to be willing to provide enough money to have a design that ourselves and our review committee is satisfied with.

We will not only be buying all of the final components for our project down the line but we will also most likely be buying parts in between that we do not end up using for the final device. For example if we buy a speaker but decide we do not like the sound quality of it or if when the screen arrives we decide that it is actually not the correct size that we would like we will need to order more parts. We will also have components that we only use for testing purposes. Our printed circuit board will include the ATmega328P microprocessor from the Raspberry Pi and so we will likely use a Raspberry Pi board for testing as it is unrealistic to design the PCB and wait for it to be shipped to us before we even test all of the components. We will also, as a group, need to cover the cost of having the PCB printed and delivered to us and possibly be able to cover this cost multiple times as we are likely going to have to redesign the PCB at least once. Another cost to keep in mind is the cost of creating the casing for our game. Fortunately one of our team members already owned a 3D printer before the start of this project so this will alleviate some of the cost for the project as we will not need to buy a 3D printer and it also makes it so we will not have to outsource to get the casing of the game built and the filament that the game will be made out of is relatively inexpensive.

The final iteration of our device is intended for mass production for consumers to use. With this in mind we are making decisions as if we intend to profit from selling our device. Of course in an outside setting where the game is being mass produced, there would be discounts for purchasing components in bulk and also there would not be expenses for research and design. Having these discounts would be necessary because in a retail setting our game would have to be as cheap as a lot of these other devices are so we could not sell it over the price of our current expenses (~$210).

4.2.3 Social Constraints

Our most realistic social constraint we will be faced with is how the user interacts with the game, more specifically how we are able to get the user to play the game correctly. In order to get the user to understand the game correctly so that they get the most out of it we will need to have a very good explanation of how the game will work. For the purpose of this project this explanation will be done in this paper, however in a theoretical retail setting to ensure that the consumer has a sound understanding of the game, we would have to provide a user's manual. In this manual we could have detailed written instructions and potentially a link the consumer could follow that would take them to a short video demonstration. In this retail scenario our group thinks it would be necessary to have both so that the consumer fully understands the game since this is a critical part of our device.

4.2.4 Equipment Constraints

For our project we do not have too many equipment constraints. Our biggest equipment constraint will be related to the software and programming language that we are going to use for our microprocessor. Since the microprocessor has a required programming language that it needs to be programmed in to understand we must use this language. For the microprocessor that we decided on, the Broadcom BCM2835 SoC, we must use a variant of C++. Specifically the Raspberry Pi which we will be prototyping on uses this language. The difference between the Broadcom BCM2835 SoC C++ and regular C++ is that it has special methods and functions added for ease of use. While the group does not have a tremendous amount of experience with C++ we all have plenty of experience with programming languages in general which greatly help our ability to translate what we know into the C++ programming language.

Another equipment constraint we may have is our 3D printer. The 3D printer that we have is a modified Taz 7. Since it is not a commercial 3D printer it has its own set of limitations. Some of which being the size of the bed, its consistency and also our own ability to print the specific shape we are going for. The size of the bed will greatly influence how we print the shell of our project. It will determine if we must print the device in individual pieces and then put them together via glue or snapping them together. If the bed's not big enough to print the entire device in one session then we will have to do one of these options. Fortunately this won't change the entire design of the device as we can make it work by just printing it piece by piece and putting them together,this will only affect how we go about printing it. We also must consider consistency with 3D printers. Since 3D printers are a newer technology they do have some errors while printing larger scale projects. Because of this we will have to plan ahead and make sure we give ourselves ample time to test different designs, filaments and tactics when printing.

4.2.5 Safety Constraints

The safety constraints of our project are to ensure the group stays safe in the process of building the device as well as being positive that any consumers of the device will be just as safe. The construction of our project will have low risk of energy. Since it is a small handheld device we do not plan on having to use any power tools, high voltages, or any dangerous devices. There is the possibility that we may need to solder some components, however we do not anticipate this as we will be getting an outside company to print our PCB and this is likely where we would need to solder. In the event that we do have to solder we will take all necessary cautions to ensure we do not endanger ourselves. The main concern when using a soldering iron is lead poisoning. If handled incorrectly lead can get transferred to the hands of the operator and if they do not wash their hands then it can potentially be ingested causing health problems. However, as previously stated we do not plan to have to use such device. For the non PCB components that we plan on soldering and any PCB components we end up soldering, one of our members has the experience and equipment to do so safely.

When it comes to ensuring the safety of our consumers, our responsibility lies in carefully protecting them from any potential hazards. One crucial aspect of this involves meticulously inspecting and addressing the electrical components of our products. We must guarantee that there are absolutely no exposed wires, and all electrical components that may be susceptible to unsafe contact are securely enclosed within our protective casing. This measure not only preserves the integrity of our devices but also mitigates the risk of accidental electrical shocks or other related mishaps.

Furthermore, it is imperative that all internal wiring is meticulously organized and precisely positioned within our device. This meticulous arrangement is essential to prevent any short circuits, which could result in severe internal issues, potentially leading to fire hazards. By upholding this standard of precision, we prioritize the safety of any potential consumers.

Additionally, we cannot understate the concern of battery safety. In order to ensure consumer well-being we must make sure that our batteries are not only correctly installed but also firmly secured in place. Mishandling or improper installation of batteries can have catastrophic consequences, such as explosions, fires, and other hazardous incidents. Hence, we leave no room for compromise when it comes to battery safety, understanding the potential dangers that lie in neglecting this aspect.

4.2.6 Environmental Constraints

Environmental considerations for this project primarily revolve around the battery selection. We are currently in the process of deciding whether to opt for a rechargeable battery that can be recharged using a micro USB cable or to utilize a conventional commercial battery such as AA or AAA. Whichever choice we make, it is imperative that we remain conscientious about electronic waste. This concern extends beyond the battery and encompasses our entire research and development phase. It is essential to avoid unnecessary purchases of excess parts, both due to budget constraints and environmental responsibility. Wastefully acquiring components not only strains our financial resources but also contributes to unnecessary energy consumption in their production, which, if unused, results in eventual disposal.

Selecting the rechargeable battery option aligns with our commitment to reducing electronic waste, as it promotes reuse. However, the final decision hinges on what best suits our product's requirements. To manage excess waste for other components, our approach involves striving to return unused parts or, at the very least, ensuring they are recycled. Implementing such responsible disposal practices not only minimizes our environmental footprint but also potentially enhances our funding prospects, making it a win-win strategy in our project's sustainability and financial stability.

4.2.7 Ethical Constraints

Our project does not plan to have many ethical constraints since we will not be asking for any user information. A common concern with all electronic devices is how they are monitoring you. We have seen this time and time again as CEOs are on trial for the misuse of consumer information. We will not need to worry about this as our device is entirely off the grid, there will be no network or internet connection and there will also be no internal microphone for our device.

The only ethical constraint that may be relevant to our device is the previously stated safety constraints. As long as we follow the aforementioned safety constraints we will be in good shape to avoid any ethical constraints for the duration of our project even if it were to be commercialized.

4.2.8 Political Constraints

Our project will not have any political constraints. As mentioned before our product will not be connected to any sort of internet or other network meaning that there is no possibility for it to collect any user data without consent. At the time of writing this paper, there are currently no such restrictions for creating basic personal projects like this one. If we were to hypothetically commercialize this project, then we would need to do more research to determine if there were more policies to abide by. However, given the project's current scope and design, it falls within the established legal and political framework, thus eliminating potential political constraints at this stage.

4.2.9 Legal Constraints

Similar to our section on political constraints we will not have any foreseeable legal constraints either. The only possibility would be any licensing issues if we were to hypothetically commercialize our product. There could potentially be copyrights or licenses on similar products that forbid the design of ours however since we did not fully intend to sell this commercially we have not researched this. However, if that perspective were to change we would most definitely look into such issues but it is unlikely that there are any of these copyrights on our device since it is one of a kind.

4.2.10 Manufacturability Constraints

The manufacturing constraints of our device will mainly come from total cost, accessibility of parts and also our own ability to assemble the device. Our project will likely cost no more than $200 - $250 which is a great price point to be at since we are all unable to provide too much into this project being college students. This means in theory if we were to sell the device we would initially have to charge over this amount which we think is excessive for the type of device it is. However, previously stated in the economic constraints section there are generally deals for mass producing these types of products. If we were to sell this device commercially we would ideally find a way to bring the price of production to less then $40 with the idea of marketing the game at $50 a unit. If we were to put our device on the market at the previously stated price we would not have a very competitive price so it would be mandatory to bring the cost of production down.

4.2.11 Sustainability Constraints

We aim to create a long-lasting and durable device, we are committed to ensuring that it remains sustainable for as long as its parts permit. Our primary goal is to design a device capable of withstanding moderate wear and tear, even during extensive gameplay sessions. To achieve this, we will place a strong emphasis on the build quality, striving to attain the highest possible standards given the materials we have.

A critical factor in determining the device's robustness is the type of filament we will use for the 3D printer. This filament choice directly influences the device's ability to handle the previous mentioned wear and tear. While we may have limited control over the inherent quality of individual components, we have the ability to select components of higher quality, thus contributing to the overall durability and performance of the device. It's important to note that our device is not designed for extreme weather conditions. To make this information abundantly clear to hypothetical consumers, we would consider including a warning label or a small handbook in the packaging if we were to bring the device to the commercial market. This precautionary measure is intended to enhance the device's longevity and ensure that users are well-informed about its intended usage parameters.

4.2.12 Constraints Summary

To summarize, there will be a number of constraints that will be providing some guidance for how we will research and design our project. Our main constraints are without a doubt time and money constraints. We will have to be adamant about following these guidelines in order to succeed in this project. The next most important constraints we will be abiding by are the safety constraints as we believe this to be almost just as important. The reason this is not higher up on our priority is because only a select few people will be handling the device in its entire lifespan so its overall safety does not need to be up to the same standards as a commercial product.. Also, as stated before the production of the device will not be inherently dangerous as we are using low voltage devices and will not be using any power tools. Our next most important constraints would most likely be social and environmental. As for other constraints such as legal or political we do not anticipate having many if any constraints for these.
To restate again the time constraint will be of the utmost importance to the group as this entire project will be the main factor for if any of us graduate in spring 2024. With this in mind we must remember the due date for this 45 page paper, the dude date for our 90 page paper at the end of this semester and then of course our demonstration at the end of next semester in spring. With all of this in mind we will continue to use our time wisely and give ourselves as much time as possible to order and test components as this is the largest constraint.

For our safety constraints we know that we must minimize risks by avoiding power tools and high voltages in the construction of our small handheld device, with potential soldering precautions in place. When soldering is necessary, we will take measures to prevent lead poisoning. For consumer safety, we meticulously protect against electrical hazards by enclosing components and ensuring organized wiring to prevent short circuits. Battery safety is also a top priority, with a focus on proper installation to prevent dangerous incidents like explosions or fires. Our commitment to safety extends to both our team and potential consumers.

Environmental constraints for this project mainly center on battery selection, with the choice between rechargeable and conventional batteries. The focus is on reducing electronic waste throughout the research and development phase by avoiding unnecessary component purchases, aligning with budget constraints and environmental responsibility. The aim is to reduce waste, promote reuse with rechargeable batteries, and implement responsible disposal practices to enhance sustainability and financial stability. Our main social constraint is ensuring users understand and engage with the game correctly. In a retail context, we would provide a detailed user's manual, including written instructions and a video demonstration, to ensure consumers' comprehensive understanding of the game, which is crucial for our device's success.

Over the duration of this constraints section we have laid out many guidelines for ourselves. We plan to use these constraints in order to allow us to make smart and informed decisions regarding the research and design of this project. It has proven to be important to have researched and displayed all of the constraints for a project this large. It will provide a resource that we can come back to over the next two semesters and will continue to guide us this whole time.

Chapter 6: Hardware Design

6.1 General Design

During the process of creating the design for our project we have had multiple iterations. There were many things taken into consideration when thinking of the project design. Some of which being weight, size, comfort, and part availability. When originally thinking of the design we wanted to have the screen curve as stated above. We quickly realized that this would not be feasible for us. In order to get a curved screen we would need to have it specially made for us and this proved to be too expensive. We then started to think of the alternate methods stated above.

6.1.1 Shape and size

We then shifted our focus to how large this device would be. We have to take into account fitting all of the electrical components inside such as the PCB, MCU, and wires. We also need room to have the speaker and analog stick while leaving space for a possible haptic feedback device in the base of the controller. We are not entirely sure at the moment what size the device will be in order to incorporate all of the previously mentioned specifications however we have seen many controllers that contain all of these components plus more that are about 90mm x 120mm. We hope to have our final design in this range with the expectation that our controller may be slightly larger because of our lack of professional equipment.

Our goal is to 3d print the casing for the controller in order to cut costs as much as possible while also providing a lightweight shell. This will allow us to test out different designs so we can test different comfort levels as well as allowing us to mold the controller to the exact dimensions we need to provide enough room for all the components previously stated. This will also allow us to test completely different shapes for the controllers.

Our team has discussed doing a single handle for the device with the display being on top of it. We have also discussed a single controller with the display to the side of the device and the last idea is having Two controllers for the device, one on each side of the device allowing for both hands on the device to maximize control. Each iteration would have its pros and cons of course. For the single handle with the screen on top, this would allow for a sleek one handed design with the display directly in front of the user. However, we would have to consider counter balancing the device so that way it is not top heavy. For the design where the screen is to the side of the screen this would be a more traditional look when compared to other video game devices today, however we would then need to worry about the weight of the screen causing a rotational force to the direction of the screen which may be difficult for the user to control depending on the weight of the screen. Finally the design where the device has two handles on each side of the display would allow for the best balance of the device, however currently, only one side of the device would have any controls so this may cause some confusion with a user.

6.1.2 Weight

As for the weight of the device this will heavily depend on the screen choice that we end up going with. We estimate that most of the weight will come from the weight of the screen since the other components are fairly lightweight. We will have to make sure that the device is not top heavy so there may be a need to add some counterweights to the controller in order for the device to feel correct in the user's hand. The estimated weight in our engineering specifications is about 600 grams. This weight is based on the Valve Index controller. The weight of this controller is 196 grams however we estimate that the screen will of course add additional weight to this.

6.2 Case

The case design went through multiple iterations as we advanced the idea of our project.

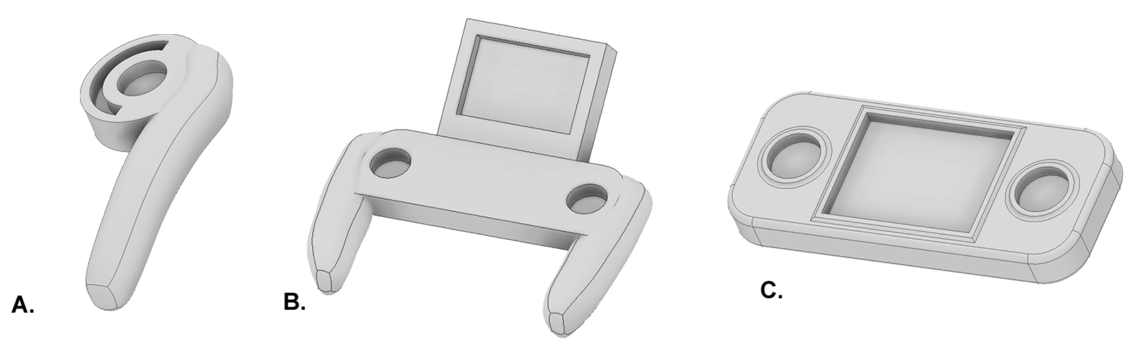

Figure 12 - Case Design Iterations

The initial designs A and B were deemed too difficult due to the strange shape of the screen in A and the unnecessarily hard task of 3D printing design B. We decided on design C because it fit our final idea of how the game would play and holds all the components necessary while also being relatively simple to print and put together. The case will be printed in three pieces; a front, back, and battery cover. The front and back will be secured together using threaded inserts that are melted into the plastic using a soldering iron, and screws coming through the opposite piece.

In addition to the shape of the case, the material we use to print the case was also a consideration. Several materials were considered; PLA, ABS, and ESD-Safe PLA. We decided on using ABS due to its smooth feel and durability. PLA is cheaper but melts at a much lower temperature making it structurally inferior, and ESD-Safety isn't really a concern for our design

Chapter 9: Testing

9.1 Simple Components

A number of the components that we have chosen can be tested with little to no programming to see if they will work for this project. This includes the haptic feedback and the toggles. As a group we already have experience communicating with toggles using microcontrollers, so it's only a matter of checking if the toggle is precise enough for the eight sections of the game and consistent enough to not frustrate the player. The haptic feedback is even simpler, requiring only a few lines of code to run it in a way that would simulate how it will feel for the user. While we can't test the haptic feedback with a finished product, we can estimate the weight and use a similarly sized object to feel how the haptic feedback can be expected to perform inside the case of our final design.

9.2 Involved Components

There are a number of components that will require a level of programming that is pretty close to our end goal to properly test; this includes the microcontroller, the speaker, and the LCD and its driver board. These all require a pretty developed version of our final code because the performance of the microcontroller and LCD driver running the game needs to be tested to ensure that the game can consistently run at a high enough frames per second to be enjoyable without burning any of the components. We may be able to find demos online to prove that the LCD and its driver can run at the frames per second and resolution that is advertised, but to truly test them we will need the game logic to be running at the same time. The speaker requires less code as we can test any sound effects we want before having a playable game.

9.3 Final Components

The final components that we can test are the case and the battery. Both of these components require an almost entirely finished design before they can be tested. The case has to fit every component in a way that is rugged enough to not risk breaking anything, and thus the sizes of everything must already be known. The battery as well requires all the components running at game speed to truly be tested.

9.4 Main Menu

When the system is first powered on the user will see the main menu, from here they can change difficulty settings, see the leaderboards, and start the game. The menus are navigated using the right toggle and selections are made using L3 (clicking the left toggle.) The leaderboards section will allow the user to scroll through the top one hundred recorded scores and see the initials that the player entered for that score.

The difficulty settings that can be changed are the speed at which the game plays and if the user wants the arrows on the sections to slowly fade away, adding a memory element to the game. Both of these modifiers will have a multiplier effect on the score. The other setting is if the user wants to use the left, right, or both toggles. Each toggle option will have its own scoreboard in the leaderboards section.

9.5 In-Game

Once in-game the user can only input on the toggles that correspond to what mode they are in. There is no pause as repeatedly pausing and unpausing the game would bypass the difficulty. Once the user fails to clear a level and enters the game over screen, they will be asked to enter their initials using the toggles for the leaderboards. After the user enters their initials they will be sent back to the main menu.

Chapter 10 Administrative Content

Budget and Financing

Item Description	Quantity	Total Estimated cost
Display	**1-2**	**$45**
RA8875 Driver board	**1**	**$40**
Toggle	**1**	**$10**
Speaker	**1**	**$10**
Controller (casing)	**1**	**$35**
Printed Circuit Board	**1**	**$50**
Microcontroller	**1**	**$20**

Total estimate		**~ $210**

Table 10 - Estimation of total product cost

Project Milestone

Task	Complete by date	Status
Project discussions	August 25th, 2023	Complete
Project Decision	September 4th, 2023	Complete
10 page Divide-and-Conquer report	September 15, 2023	Complete
Component research and decision	September 28th, 2023	Complete
PCB research	October 12th, 2023	In development
Bill of Materials	October 20th, 2023	Complete
Table of Contents	October 28th, 2023	In development
45 Page Documentation	November 3rd, 2023	Complete
Final 90 Page Documentation	December 5th, 2023	In development
Obtain parts	During winter break starting December 2nd, 2023	In development
Senior Design 2	----------------------------------	-----------------------------------
Individual part testing	N/A	N/A
Assembly	N/A	N/A
Full testing and redesigning	N/A	N/A
Final testing	N/A	N/A
Presentation	N/A	N/A

Table 11

Appendix A - Sources and references

[1] “Home.” *Arduino*, www.arduino.cc/. Accessed 5 Oct. 2023.

[2] “IEC.” *IEC 61508:2010 CMV | IEC Webstore*, webstore.iec.ch/publication/22273. Accessed 20 Oct. 2023.

[3] JayconSystems, and Instructables. “Getting Started with the ATMega328P.” *Instructables*, Instructables, 29 Sept. 2017, www.instructables.com/Getting-Started-With-the-ATMega328P/.

[4] “Joysticks.” *Joystick Information and Circuits*, www.epanorama.net/documents/joystick/ataristick.html. Accessed 15 Oct. 2023.

[5] *Lead Soldering Safety*, blink.ucsd.edu/safety/occupational/hazard-control/lead-soldering.html#:~:text=Workers%20can%20be%20exposed%20to,and%20muscle%20and%20joint%20pain. Accessed 29 Oct. 2023.

[6] Toy Industry Association, Inc. “U.S. Sales Data.” *The Toy Association*, www.toyassociation.org/ta/research/data/u-s-sales-data/toys/research-and-data/data/us-sales-data.aspx. Accessed 28 Sept. 2023.

[7]“USB.” *Encyclopædia Britannica*, Encyclopædia Britannica, inc., 13 Sept. 2023, www.britannica.com/technology/USB.

[8] “MSP430FR69891.” *MSP430FR69891 Data Sheet, Product Information and Support | TI.Com*, Texas Instruments, www.ti.com/product/MSP430FR69891. Accessed 3 Nov. 2023.

[9] “ESP32-WROOM-32-N4 Espressif Systems | RF and Wireless | DigiKey.” *DigiKey* www.digikey.com/en/products/detail/espressif-systems/ESP32-WROOM-32-N4/8544298. Accessed 3 Nov. 2023.

[10] “Buy A Raspberry Pi Zero W.” *Raspberry Pi*, Raspberry Pi, www.raspberrypi.com/products/raspberry-pi-zero-w/. Accessed 20 Oct. 2023.

[11] “ATMEGA328P - Microchip Technology.” *MICROCHIP*, Microchip, www.microchip.com/en-us/product/ATMEGA328P. Accessed 23 Oct. 2023.

[12]“International Electrtechnical Commission Homepage.” *IEC*, International Electrtechnical Commission, Accessed 1 Nov. 2023, iec.ch/.