

***Physical Immersive Video Gaming:
Using Pac-man to Help Fight Obesity***

by

Mathew Buckman

Project submitted in partial fulfillment of the requirements for the
degree of Master of Science in Information Technology

Rochester Institute of Technology

***B. Thomas Golisano College
of
Computing and Information Sciences***

Department of Information Sciences and Technologies

November 24, 2014

Rochester Institute of Technology

***B. Thomas Golisano College
of
Computing and Information Sciences***

Master of Science in Information Technology

Project Approval Form

Student Name: Mathew Buckman

Project Title: Physical Immersive Video Gaming:
Using Pac-man to Help Fight Obesity

Project Committee

Name

Signature

Date

W. Michelle Harris
Chair

Elouise Oyzon
Committee Member

ABSTRACT

Video games have become commonplace in our lives today, creating a great source of entertainment but also a contributor to the growing rates of obesity. Creating an immersive 3D video game which functions with a tangible interface can help promote physical activity and deter these growing numbers. In addition, it taps into a new realm of video game which is more enveloping and exciting than traditional video games.

CONTENTS

Abstract	iii
List of Figures	vi
1. Introduction	1
2. Review of the Literature	2
3. Example Applications	5
4. Methodology	6
4.1. Proposed Methodology.....	6
4.2. Proposed Timeline.....	7
4.3. Revised Methodology	8
4.4. Revised Timeline	10
5. Implementation	12
5.1. Input Device	12
5.1.1. Microcontroller.....	12
5.1.2. Compass Sensor.....	14
5.1.3. Accelerometer	15
5.1.4. Bluetooth Modem	17
5.1.5. Putting it All Together	19
5.2. Projection Cube	23
5.3. Software	25
5.3.1. Game Engine.....	25
5.3.2. Interfacing with the Input Device.....	26
5.3.3. Environment	27
5.3.4. Player	28
5.3.5. Ghosts.....	29
5.3.6. Gameplay.....	31
6. Product Comparison	32
6.1. Accelerometer Tracking	32
6.2. Camera Tracking.....	34
6.3. Virtual Reality Headsets	35
7. Conclusions.....	39

7.1. Lessons Learned	39
7.2. Future Work	41
7.3. Conclusion	42
8. Bibliography	43

LIST OF FIGURES

Figure 1. Arduino Diecimila Components	13
Figure 2. Arduino Battery Power	13
Figure 3. HMC6352 Digital Compass Module	14
Figure 4. HMC6352 Digital Compass Module Sketch	14
Figure 5. HMC6352 Digital Compass Module Schematic	15
Figure 6. Memsic Dual-axis Accelerometer Operation	16
Figure 7. Memsic Dual-axis Accelerometer Sketch.....	16
Figure 8. Memsic Dual-axis Accelerometer Schematic.....	17
Figure 9. BlueSMiRF Gold Bluetooth Modem Sketch	18
Figure 10. BlueSMiRF Gold Bluetooth Modem Schematic	19
Figure 11. Input Device Sketch	21
Figure 12. Input Device Schematic.....	21
Figure 13. Input Device Assembled	22
Figure 14. Input Device Assembled Angled View	22
Figure 15. Input Device Attached to Belt.....	23
Figure 16. Projection Cube.....	24
Figure 17. Screenshot of Unity Editor	25
Figure 18. Labyrinth Layout	27
Figure 19. 360 Degree Camera View	28
Figure 20. Waypoints and Navigation Path	30
Figure 21. Waypoint Grid.....	30
Figure 22. Level Specifications.....	32
Figure 23. Nintendo Wii Remote	33
Figure 24. PlayStation Camera and Xbox Kinect.....	35
Figure 25. Oculus Development Kit	37
Figure 26. Oculus Rift Inside View	37
Figure 27. Virtuix Omni	38
Figure 28. Virtuix Omni in Use	38

1. INTRODUCTION

With the massive flood of video games into today's culture, sedentary activities have taken the place of physical ones in the lives of both children and adults. In 2011-2012 in the United States, 16.9% of children 2-19 years old were obese. A staggering 34.9% of adults were obese as well (Ogden, 2014). Video games and television are leading contributors to a sedentary lifestyle, with studies showing that each hour spent on these activities may double a child's risk of obesity (Warner, 2004).

Tangible interfaces allow for the real-time control of virtual characters and environments through natural, human motions and gestures. When creating an immersive 3D game or environment, interfaces designed for 2D environments are often difficult to use and have a high learning curve. By combining the digital with the physical, "these interfaces can more easily fit into the way we engage with our everyday physical environment using our body and hands than traditional computer input devices such as mice and keyboards" (Mazalek & Nitsche, 2007). Physical interfaces have been shown to be a more intuitive (Beckhaus, Blom & Haringer, 2005) form of interaction.

By combining tangible interfaces with immersive 3D games, a new domain of game play emerges which promotes physical rather than sedentary activity. Given the popularity of video games, designing a fun and challenging game for this type of environment could help combat the growing obesity epidemic by promoting daily physical activity. In addition, these types of games could prove to be more challenging

and fun than traditional non-immersive games, bringing users into the environment and moving away from the “god’s eye” view of many traditional games.

2. REVIEW OF THE LITERATURE

Baranowski, T. (2013, June 11). Games and Childhood Obesity. In *Games for Health Journal*. Retrieved November 26, 2014, from <http://online.liebertpub.com/doi/full/10.1089/g4h.2013.1502>

Obesity is a common health problem among children and adults around the world. 50% of U.S. students in lower income middle schools with a higher population of ethnic minorities are overweight or obese. There is ample evidence to show that a properly motivated player can get a moderate workout from active video games.

Beckhaus, S., Blom, K. J., & Haringer, M. (2005). *Intuitive, Hands-free Travel Interfaces for Virtual Environments*. Retrieved May 8, 2008, from University of Hamburg Web site: <http://imve.informatik.uni-hamburg.de/files/19-IntuitiveTravel-VR2005-BeckhausBlomHaringer.pdf>

Two tangible interfaces have been developed for travel through virtual worlds; a dance pad and a chair-based system. These interfaces, particularly the chair-based system, proved to be very intuitive and easy to use in virtual environments. In addition to its ease of use, the chair-based interface was found to be more enjoyable to use than traditional mouse and keyboard systems.

Dyer, O. (2007, March 15). Can Video Games Zap Childhood — and Adult — Obesity? In *National Review of Medicine*. Retrieved May 8, 2008, from http://www.nationalreviewofmedicine.com/issue/2007/03_15/4_advances_medicine_5.html

The physical nature of the Nintendo Wii system has shown much potential for aiding in the loss of weight in healthy players. In addition to weight loss, it has also shown benefits when used in rehabilitation of children with certain medical conditions.

Goldfield, G., Cameron, J., Chaput, J. (2014, March 11). Is Exergaming a Viable Tool in the Fight against Childhood Obesity? In *Journal of Obesity*. Retrieved November 26, 2014, from <http://downloads.hindawi.com/journals/job/aip/304521.pdf>

Evidence supports the use of exergaming to increase energy expenditure. Seated video games increase food intake, therefore exergaming should be encouraged and may help achieve recommended physical activity levels for children worldwide.

Gudmundsen, J. (2007, October 8). Video Game Tackles Childhood Obesity. In *ABC News*. Retrieved May 8, 2008, from <http://abcnews.go.com/Technology/Story?id=3695822&page=1>

“Video games are embedded in youth culture and are an effective tool to educate children in an interactive and entertaining way” (Gudmundsen, 2007). The video game, *The Incredible Adventures of the Amazing Food Detective*, uses this idea to teach children to live healthier lifestyles in a fun and entertaining way. While not directly affecting their physical activity, this shows that video games can successfully be used to educate children, promoting physical health and helping to reduce childhood obesity.

Maddison, R. (2011, May 11). Effects of active video games on body composition: a randomized controlled trial. In *The American Journal of Clinical Nutrition*. Retrieved November 26, 2014, from <http://ajcn.nutrition.org/content/94/1/156.short>

Playing video games and watching television have been implicated as contributors to high rates of obesity. Active video games can promote physical activity and help improve body composition. A 6 month study concluded that active video games have a small but definite effect on body composition in overweight children.

Marshall, P., Rogers, Y., & Hornecker, E. (2007). *Are Tangible Interfaces Really Any Better Than Other Kinds of Interfaces?* Retrieved May 8, 2008, from Open University, Pervasive Interaction Lab Web site: <http://www.cl.cam.ac.uk/conference/tangibleinterfaces/TUIworkshop-Marshall.pdf>

While it seems logical that tangible interfaces are a better form of interaction than other interfaces, not much research has actually been done to prove this. Further study and development in this area would be beneficial to determining the impact tangible interfaces have on learning, physical activity, and group participation.

Mazalek, A., & Nitsche, M. (2007). *Tangible Interfaces for Real-time 3D Virtual Environments*. Retrieved May 8, 2008, from Georgia Institute of Technology Web site: http://synlab.gatech.edu/data/papers/mazalek_ace2007_tui3d.pdf

Common 2D interfaces present challenges when used with 3D environments. 3-dimension, physical interfaces, such as a puppet or glove to capture motion and gestures, provide a much more intuitive, natural way to control 3D virtual elements.

Thompson, D. (2014, July 16). What Serious Video Games Can Offer Child Obesity Prevention. In *JMIR Serious Games 2014*. Retrieved November 26, 2014, from <http://doi.org/10.2196/games.3480>

Childhood obesity is a worldwide public health issue. Obese children are more likely to become obese adults, further worsening the situation. 97% of 12-17

year olds play video games, and serious video games show promise in promoting healthy behaviors to youth.

Torgan, C. (2002, June). Childhood Obesity on the Rise. In *The NIH Word on Health*.

Retrieved May 8, 2008, from <http://www.nih.gov/news/WordonHealth/jun2002/childhoodobesity.htm>

Physical activity has decreased and body weight has increased in children. Childhood obesity is now an epidemic in the U.S. with the number of overweight children more than doubling over the last few decades. As a result, obesity-related health problems have also drastically risen. Encouraging everyday physical activity is vital in reducing the problem.

Tse, E., Greenberg, S., Shen, C., & Forlines, C. (2006, May 7). Multimodal Multiplayer Tabletop Gaming. In *Third International Workshop on Pervasive Gaming Applications*.

Retrieved May 8, 2008, from <http://www.merl.com/reports/docs/TR2006-009.pdf>

Physical interfaces promote human interaction much more than do common household gaming consoles. With physical games, users are afforded the ability to remain engaged even when currently inactive in the game play. A tabletop tangible interface was developed for use in the game play of Warcraft III and the Sims, which proved both a more natural method of interaction as well as a way to turn single-player games into multi-player activities.

Warner, J. (2004, July 2). Video Games, TV Double Childhood Obesity Risk. In *WebMD Medical News*.

Retrieved May 8, 2008, from <http://children.webmd.com/news/20040702/video-games-tv-double-childhood-obesity-risk>

Studies suggest that a child's risk of obesity may be doubled for every hour they spend playing video games or watching television. A study in Switzerland showed that video game play and television watching were contributing factors to the rate of childhood obesity, while physical activity helped to lower these rates.

Wisneski, C., Orbanes, J., & Ishii, H. (1998, April 18). *PingPongPlus: Augmentation and Transformation of Athletic Interpersonal Interaction*. Retrieved May 8, 2008, from

MIT Media Laboratory Web site: http://tangible.media.mit.edu/content/papers/pdf/PingPongPlus_CHI98.pdf

Developers at MIT have created a digitally enhanced version of the game ping-pong. Using a reactive tabletop with various sensors, the game is made more interesting by incorporating video projections and sound. This coupling of digital technology and athletic activity is a good example of how the physical and virtual realms can be merged to create a more enriching experience.

3. EXAMPLE APPLICATIONS

Since publication of the articles reviewed in the previous section, several applications have been developed which implement or provide examples of the ideas discussed. This section explores some of those applications featuring tangible interfaces and active play.

One example application is the video game *Bounden* published by Game Oven in collaboration with the Dutch National Ballet (<http://www.playbounden.com>). To play the game, two individuals hold either end of a smartphone and tilt the device around a virtual sphere to follow a path of rings. Players swing their arms, twist their bodies, and perform movements which simulate dancing.

A second example is *Johann Sebastian Joust* created by Die Gute Fabrik (<http://www.jsjoust.com>). The game is played on Sony PlayStation 3 or 4 and makes use of the PlayStation's motion controller wands. The object of the game is to jostle the controller of your opponents while protecting your own. Moving your controller beyond the allowable threshold eliminates you from the game. When the music is played slowly the controllers are very sensitive to motion. When the music speeds up the threshold becomes less strict and players are given the opportunity quickly go after their opponents.

Another example is the game *Dance Central* developed by Harmonix (<http://www.harmonixmusic.com/games/dance-central>). The game is available on Microsoft Xbox 360 and works with the Xbox Kinect device to track player movement.

During gameplay, dance moves are given on screen which the player must recreate while being track by Kinect. Players are represented on screen by one of several avatars and are scored on how well they perform the given moves.

A final example application is *Wii Street U* published by Nintendo and powered by Google. The application works by displaying 360 degree panoramic views of real locations on the Wii U GamePad. The gamepad is equipped with motion sensors and the panoramic view is moved as the user moves it around. Additionally, the application can be used in conjunction with a Wii Balance Board, allowing the user to walk in place on the board to navigate the environments displayed on the gamepad.

4. METHODOLOGY

4.1. PROPOSED METHODOLOGY

The first step of the proposed methodology was to create an immersive 3D game that is both fitting to the 4-screen “Immersatorium” cube setup, as well as the pouch interface. The “Immersatorium” consists of four screens forming a cube, onto which images are projected creating a virtual immersive environment. The pouch interface is a device, worn on the hip, containing various sensors which allow for the movement through virtual environments by walking in place. The game developed would be based on the arcade game *Pac-Man* as it is a relatively simple environment in terms of 3D modeling, the labyrinth layout fits well with the projection cube setup, and allows for easy comparison between the traditional and immersive versions. Additionally, the

need to “watch your back” in the immersive form of this game supports interaction from secondary active players.

The game was to be built in a custom game engine developed using the Microsoft DirectX framework as it is more robust than other 3D environments such as Adobe Director. The decision was made to create several environments, each having a unique layout and high quality models with photorealistic textures. This decision was counterintuitive to the one choosing to use *Pac-Man* for the simplicity of its environment, and was a large contributing factor to the changes in methodology, which will be explained in a later section.

4.2. PROPOSED TIMELINE

Given a two quarter time-frame for development, game development was to be completed in one quarter (10-12 weeks) with the second quarter allotted for usability testing and refinement of the game, interface, and immersive medium. The original work schedule for game development was as follows:

- **Week 1**
 - research into the features and game play of the original Pac-man game
 - development of features that will be incorporated into the new version of the game
- **Weeks 2-3**
 - storyboarding and mock-ups (either sketches or modeled) of 3D characters and environment
 - design and sketch board layouts and object placement
 - design and sketch virtual user interface
 - obtain/record sounds to be used in the game
- **Week 4**
 - initial development of 3D models in Maya
 - refinement and texturing of 3D models
 - development of graphics for virtual user interface

- **Week 5**
 - plan and develop layout of game classes and functions
- **Weeks 6-13**
 - build/code the game
 - build the game “shell”
 - class layouts/placeholders
 - empty 3D environment
 - setup of the screen display
 - develop splash screens and initial menus
 - add functionality to interface with the pouch
 - start building the 3D environment
 - incorporate 2D virtual user interface elements
 - continue development of remaining game elements including sound
- **Weeks 14-20**
 - test and debug the game and interface
 - fix issues and continue the testing cycle until satisfied that the game is completed
 - usability testing

4.3. REVISED METHODOLOGY

The methodology originally laid out in the Capstone proposal eventually proved to be unrealistic. The decision to build an original game engine from the ground up with high quality models and textures was overly ambitious and beyond the scope of what was required and feasible in the given timeframe. This led to the project falling very behind the outlined work schedule. As a result, the project was abandoned for some time due to outside factors, including employment and financial requirements, which wouldn't have been encountered had the project been completed in a more timely manner.

After remaining dormant for a few years, project work picked back up, but with a greatly condensed timeline. In addition to a tighter deadline, institutional changes

added the challenge of finding a useable projection cube setup that would meet my needs. In the end, a setup could not be found, and building a setup of my own was added to my growing list of tasks.

While many of the decisions remained from the original proposal, such as moving away from Adobe Director and building a game based on *Pac-Man*, some changes needed to be made to simplify the project and scope something much more realistic and feasible. The first such change was to scrap the idea of building an original game engine. This was far outside the scope of what the project set out to do, and required way too much time and additional research to complete in a timeframe expected of a Capstone project. If the project was to succeed, a pre-built game engine would have to be used.

The second change made was a reduction in the visual complexity of the game. Many weeks were spent solely working on complex models and textures just for props in the game environments. For a commercially distributed product this level of work would be acceptable. But for a prototype that kind of detail is unnecessary and becomes a hindrance to progress. Therefore, the environment would be altered to something simple, yet still visually appealing. Additionally, models would be simplified, and only required objects would be placed in the environment.

A third change was to that of the complexity of the game itself. Originally, it was intended that there would be at least three different labyrinth layouts with different difficulty settings affecting how ghosts behave. The original *Pac-Man* game was built

using just one labyrinth layout with various properties changing as a player progresses through levels, affecting the difficulty of gameplay in a more programmatic way. While it added to the complexity of the project code, implementing this method of game progression drastically reduced the amount of required time consuming tasks such as designing labyrinth layouts and placing wall models, dot models, and movement waypoints. It also expanded the number of playable game levels from three to an infinite number with over twenty levels of difficulty.

4.4. REVISED TIMELINE

Putting the project off for an extended period of time put me in a tight spot in regards to the time available to complete the project. Instead of the original 20 weeks of development time, I now had 11 weeks, with only 5 of those weeks dedicated solely to software development. Additionally, I now had the requirement of building my own projector cube setup. As a result, the following revised timeline was used to guide the project development:

- **Weeks 1-6**
 - Review status of previous work
 - Determine remaining work
 - Adjust scope and complexity of the project
 - Investigate game engine options
- **Week 7**
 - Become familiar with Unity game engine
 - Build & place environment models (walls, dots)
 - Player movement
 - Score management
 - Ghost movement
 - Heads Up Display
- **Week 8**

- Bonuses
 - Management
 - Modeling
- Levels
 - Progression
 - Difficulty
- Player health
- Tunnel areas
- Ghost release from pen
- **Week 9**
 - Frightened mode
 - Ghost phases (chase, scatter)
 - Additional modeling
 - Ghosts
 - Dot & energizer
 - Texturing
 - Walls
 - Ground plane
 - Ghosts
 - Dot & energizer
 - Bonuses
 - Game audio
 - Menu system
 - Main menu
 - Options
 - Pause menu
 - Game over screen
- **Week 10**
 - Multiple directional cameras
 - Portal texture & audio
 - Build projector cube
- **Week 11**
 - Finalize assembly of input device
 - Complete Arduino code for input device
 - Add menus for hardware interface
 - Connect to device
 - Calibrate device
 - Testing, tweaking, & bug fixing

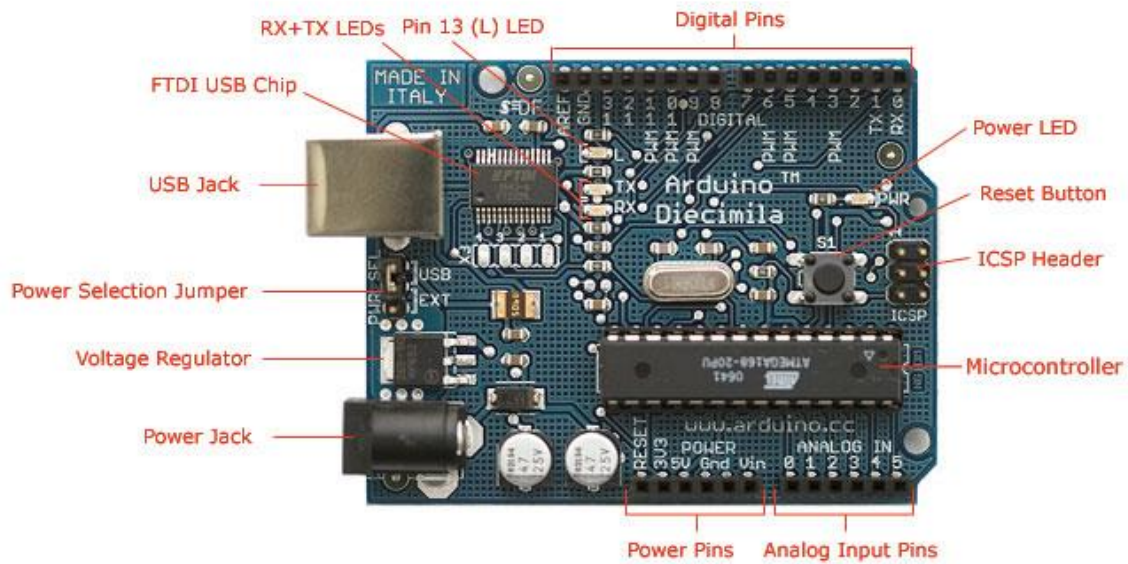
5. IMPLEMENTATION

5.1. INPUT DEVICE

5.1.1. MICROCONTROLLER

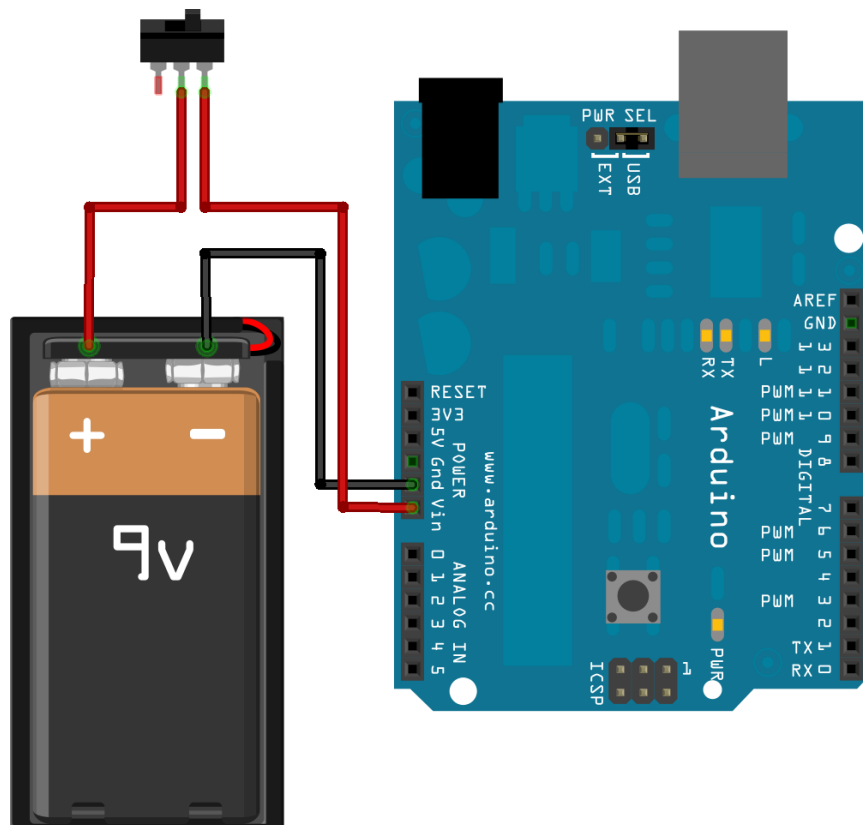
The input device makes use of the Arduino Diecimila microcontroller board. The board contains 14 digital input/output pins and 6 analog input pins. Additionally, it contains a USB jack for uploading code and powering the device, and can also be powered through its power jack or by connecting a power source directly to its ground and voltage in pins. For this project, the input device is powered by a 9 volt battery run through a toggle switch to allow it to be turned on and off, as seen in Figure 2. The microcontroller has 14 KB of useable flash memory. This is not a lot of space, so the Arduino code must be efficient and as simple as possible. The board is programmed using the Arduino programming language, which is an implementation of Wiring, a physical computing platform based on the Processing programming environment.

Figure 1. Arduino Diecimila Components



Photograph by SparkFun Electronics. Used under the Creative Commons Attribution Share-Alike 3.0 license.

Figure 2. Arduino Battery Power



5.1.2. COMPASS SENSOR

A Honeywell HMC6352 Digital Compass Module is used to determine a player's heading with an average accuracy of 2.5 degrees. The microcontroller works with the compass sensor by first sending a request for a data read and then, after a 6ms pause, requests the data from the sensor. The compass sensor calculates the new heading and reports the value back in tenths of degrees. The compass module is connected to the microcontroller board's analog 4 and 5 pins, as seen in Figure 4 and Figure 5.

Figure 3. HMC6352 Digital Compass Module



Figure 4. HMC6352 Digital Compass Module Sketch

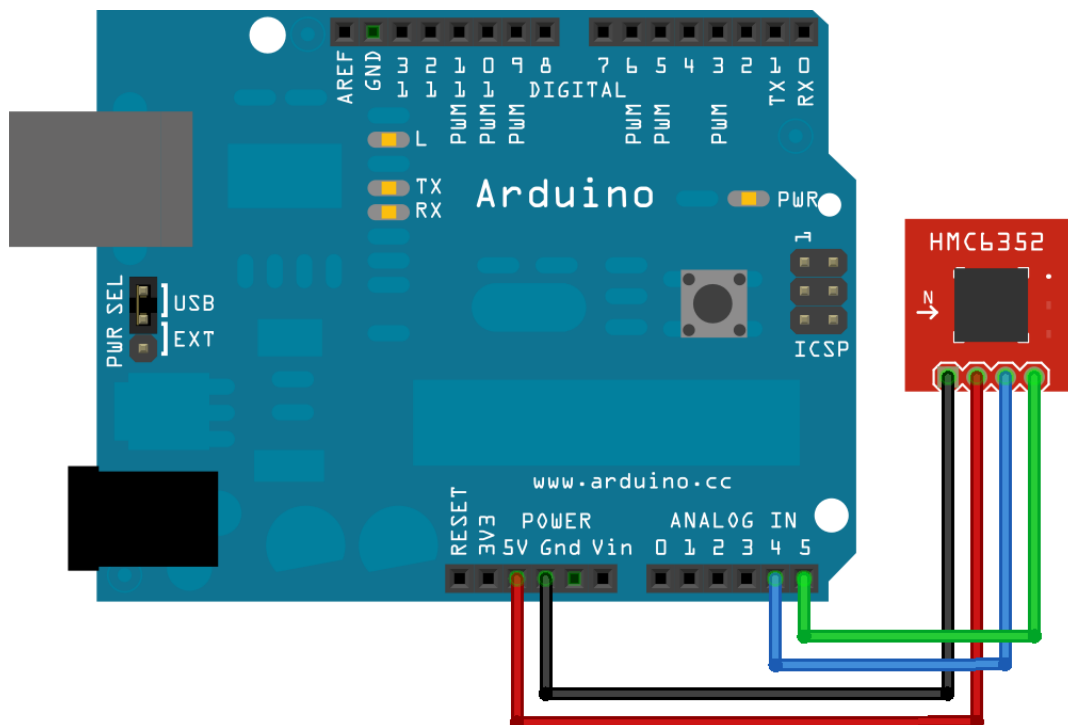
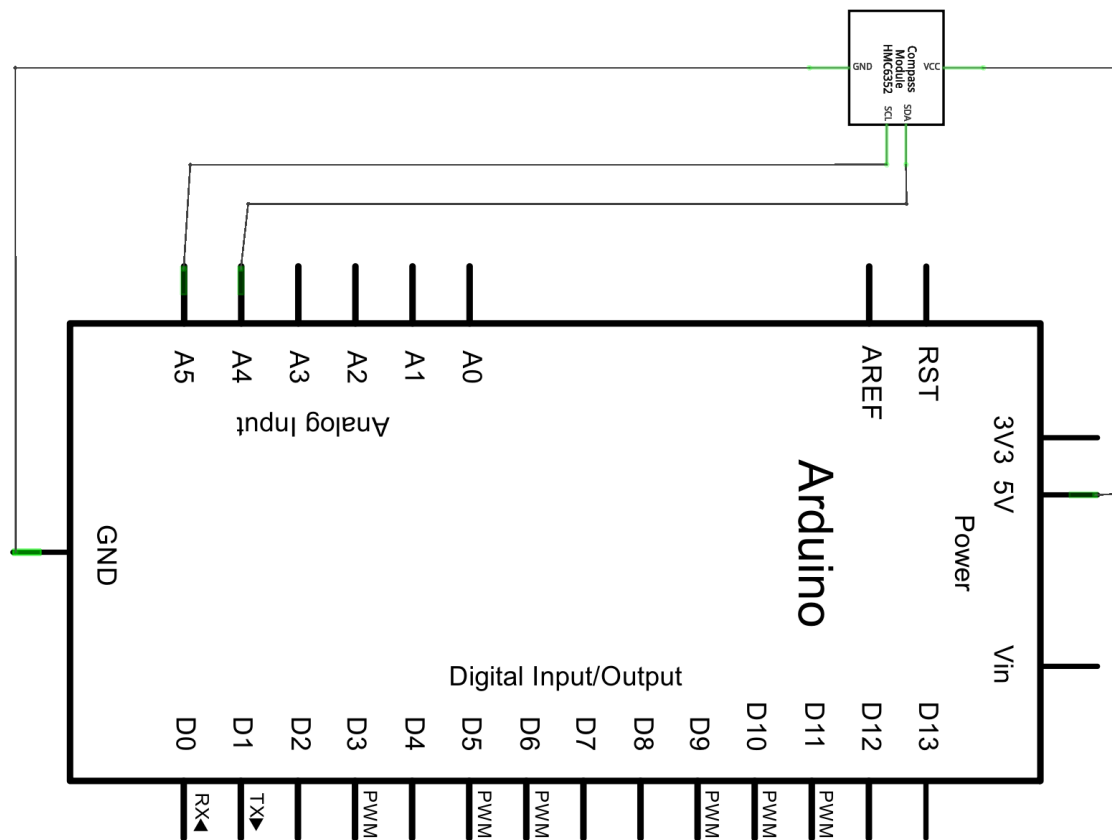


Figure 5. HMC6352 Digital Compass Module Schematic



5.1.3. ACCELEROMETER

To detect player movement, the Memsic 2125 Dual-axis Accelerometer is used.

The accelerometer operates using a chamber of gas with a heating element at the center of four temperature sensors, as seen in Figure 6. As the sensor is tilted or moved along one of its axes, the pocket of hot gas moves closer to certain temperature sensors. The sensor data is then used to determine static and dynamic acceleration. Pulses are sent out through the sensor's X and Y axis pins in durations relative to the acceleration along those axes. Those pulses are read by the microcontroller board through digital pins 4 and 5, and changes in acceleration along the Y (vertical) axis are used to determine when a player is walking in place.

Figure 6. Memsic Dual-axis Accelerometer Operation

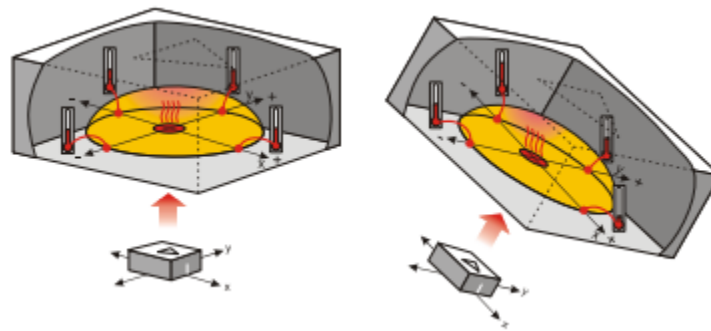


Figure 7. Memsic Dual-axis Accelerometer Sketch

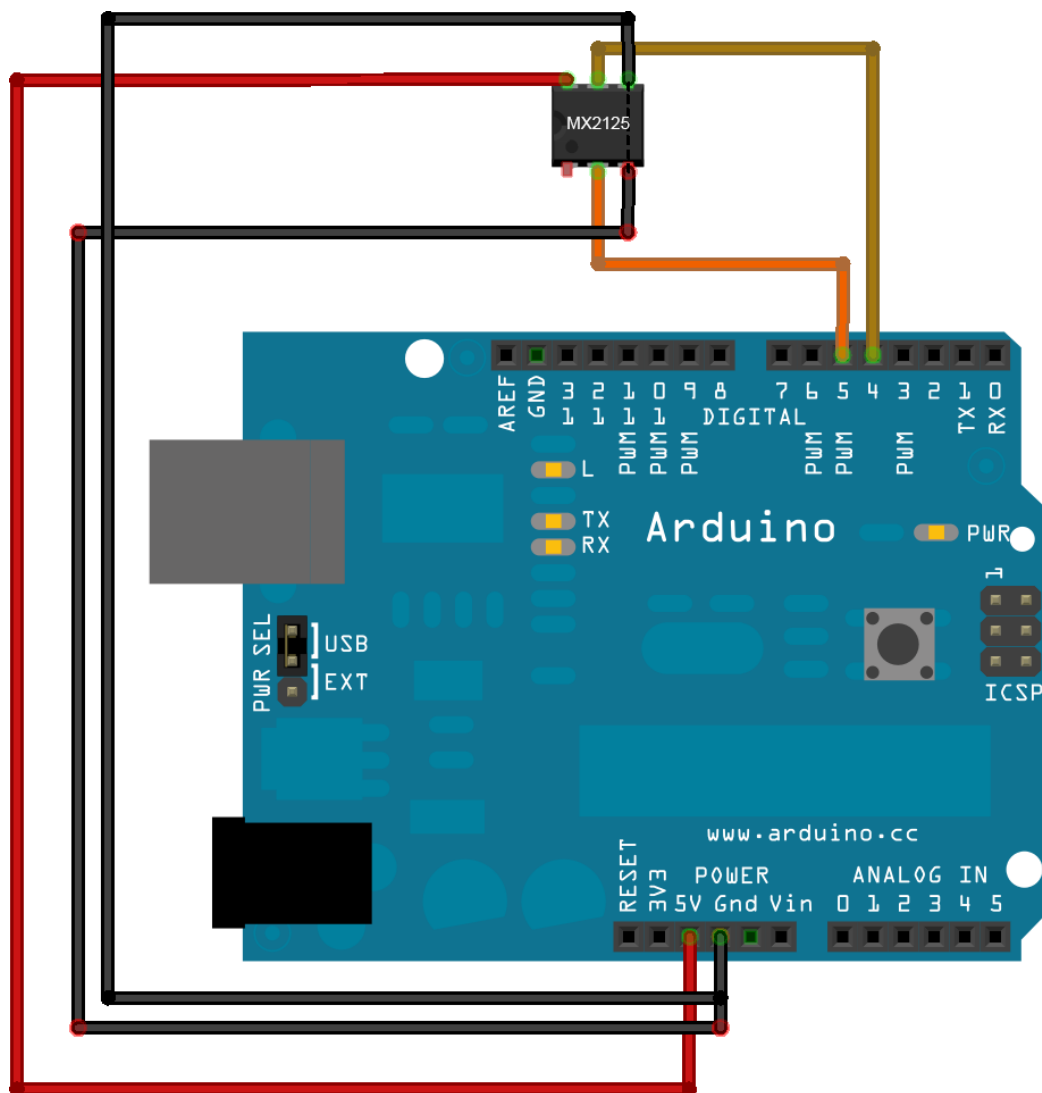
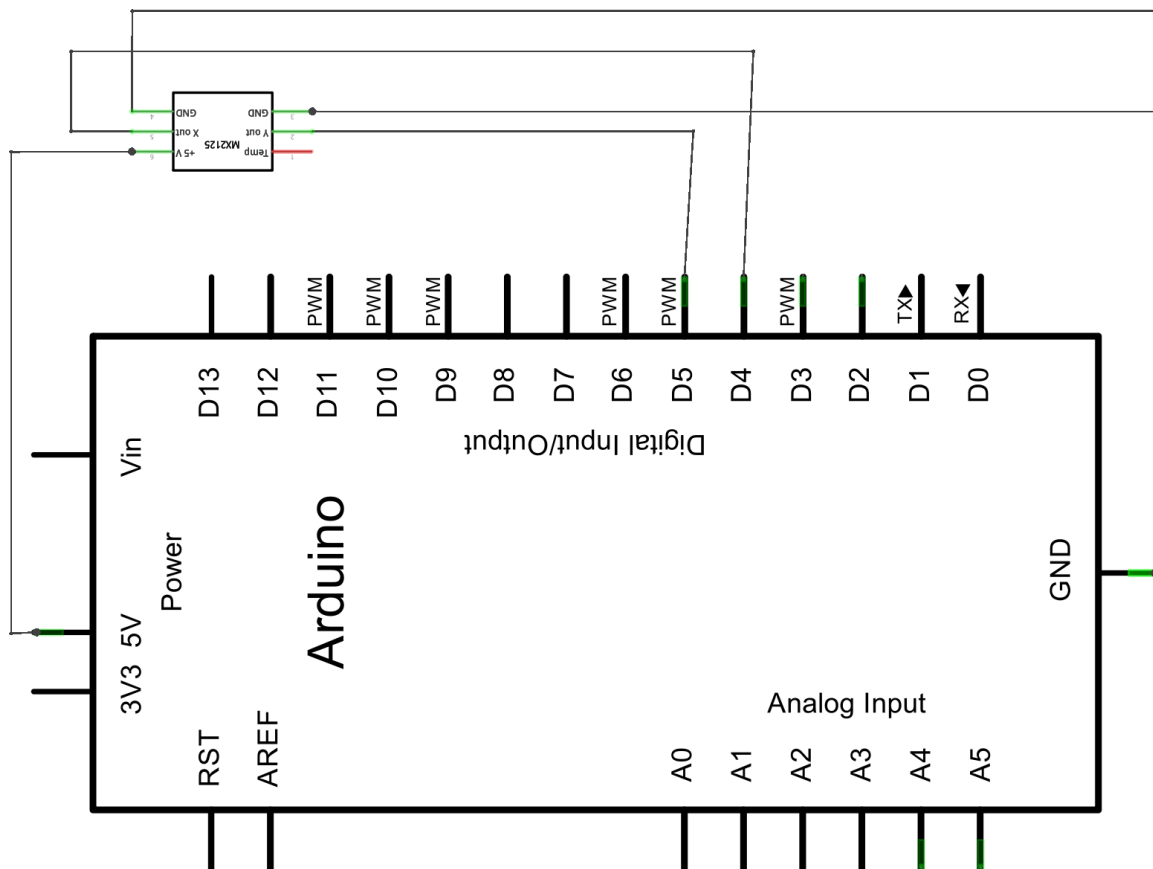


Figure 8. Memsic Dual-axis Accelerometer Schematic



5.1.4. BLUETOOTH MODEM

To facilitate wireless communication between the input device and the game, the project uses the BlueSMiRF Gold Bluetooth Modem. The modem acts as a serial pipe, continuously sending available data to, and listening for available data from the game. A Bluetooth USB dongle provides an interface to the computer running the game. Serial communication between the devices is handled with Arduino's SoftwareSerial library. The modem is capable of streaming at a rate of 115200bps, but the stream was reduced to 9600bps for this project as the SoftwareSerial library has proven to be unreliable at faster speeds. The modem has been successfully tested at a

distance of 350 feet which, while the player will likely never be more than 10-15 feet away from the machine running the game, opens up the possibility for tremendously large immersive environments.

Figure 9. BlueSMiRF Gold Bluetooth Modem Sketch

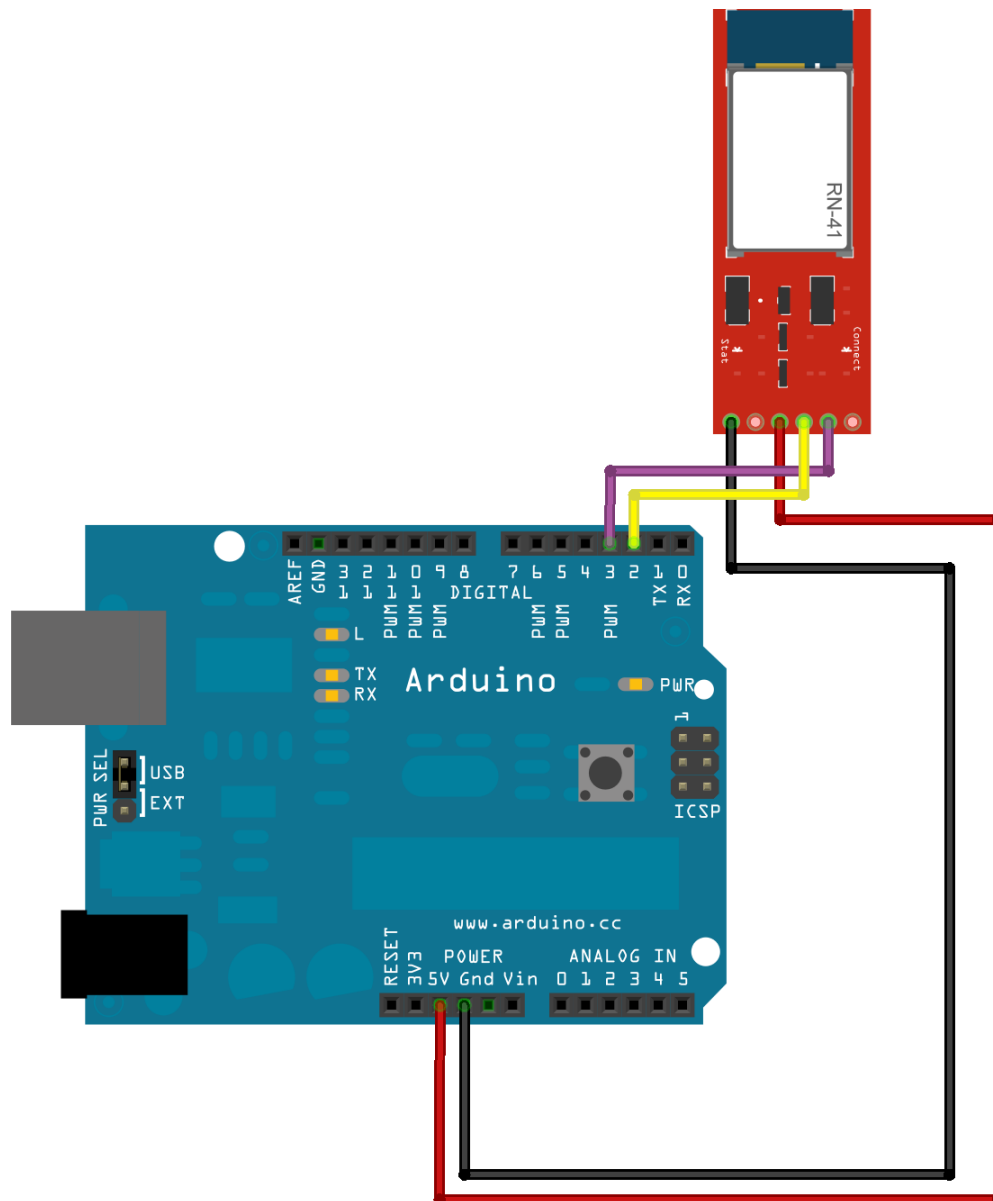
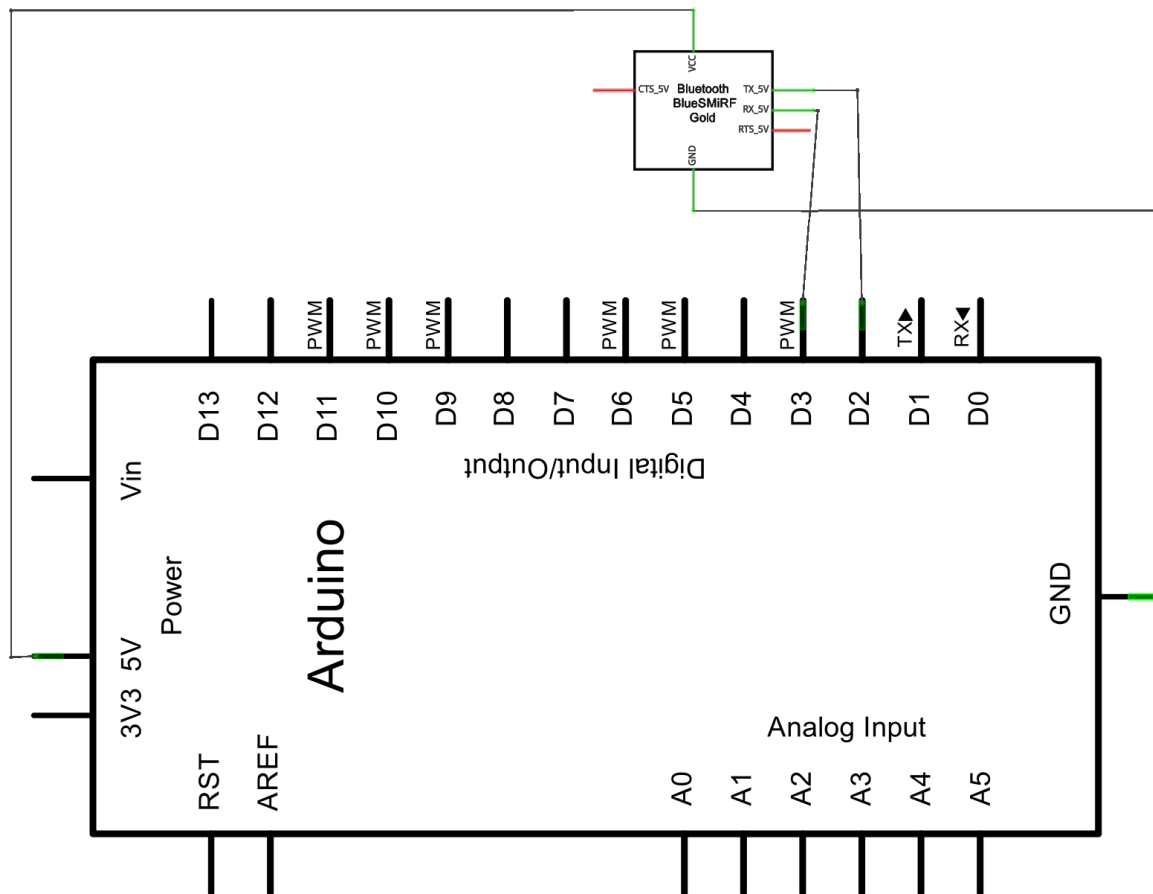


Figure 10. BlueSMiRF Gold Bluetooth Modem Schematic



5.1.5. PUTTING IT ALL TOGETHER

To combine all components of the input device, the sensors were first combined on a generic printed circuit board. The compass sensor, accelerometer, and Bluetooth modem were all soldered onto the circuit board with header pins either attached directly or through a connection on the board. Header pins were used for easy and reliable connections, making testing and adjustments much simpler. Rows of header pins were also inserted into all pin slots on the Arduino board, providing much more solid connection points and, again, simplifying testing and adjustments.

A 5x2.5x2" plastic project enclosure is used to house all components. The Arduino board was secured to the enclosure first using 10mm standoffs. After that, the printed circuit board with the additional components was secured over the Arduino board using 30mm standoffs. Once both boards were secured, jumper wires were used to connect the header pins for the individual components to the appropriate header pins in Arduino's pin slots. The final step was to connect a 9 volt battery to the Arduino board through a toggle switch and close everything up.

To allow the player to wear the input device, the enclosure is attached to an adjustable cotton belt. The belt provides a sturdy, yet adjustable way to strap the device to a player's waist. Four holes were drilled through the belt and the base of the plastic enclosure. Finish washers were placed on the inside of the belt, and a 3/8" machine screw was passed through each washer, the belt, and then the enclosure base. Another washer was placed on the inner side of the enclosure base, followed by a hex nut to secure each screw.

Figure 11. Input Device Sketch

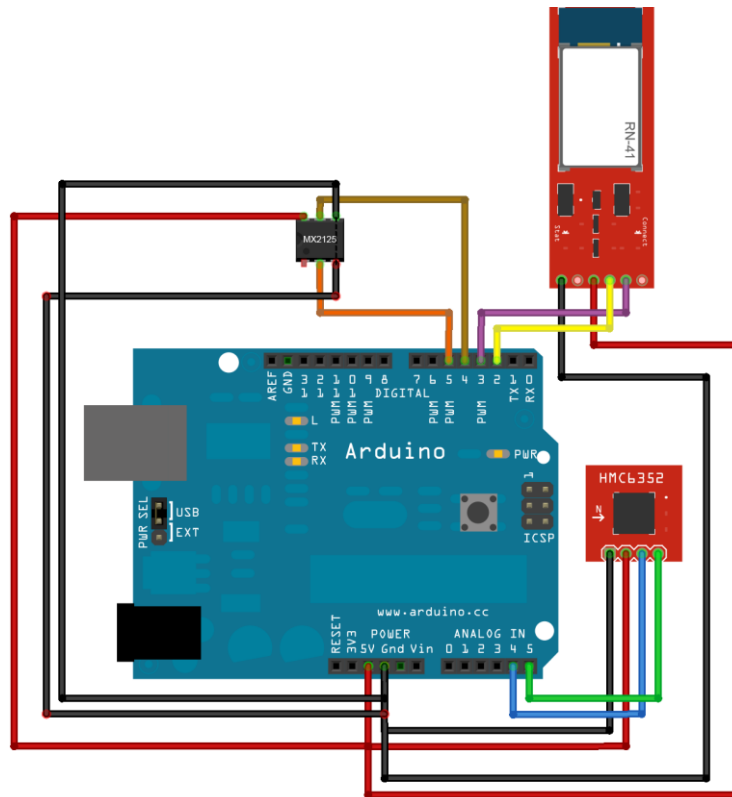


Figure 12. Input Device Schematic

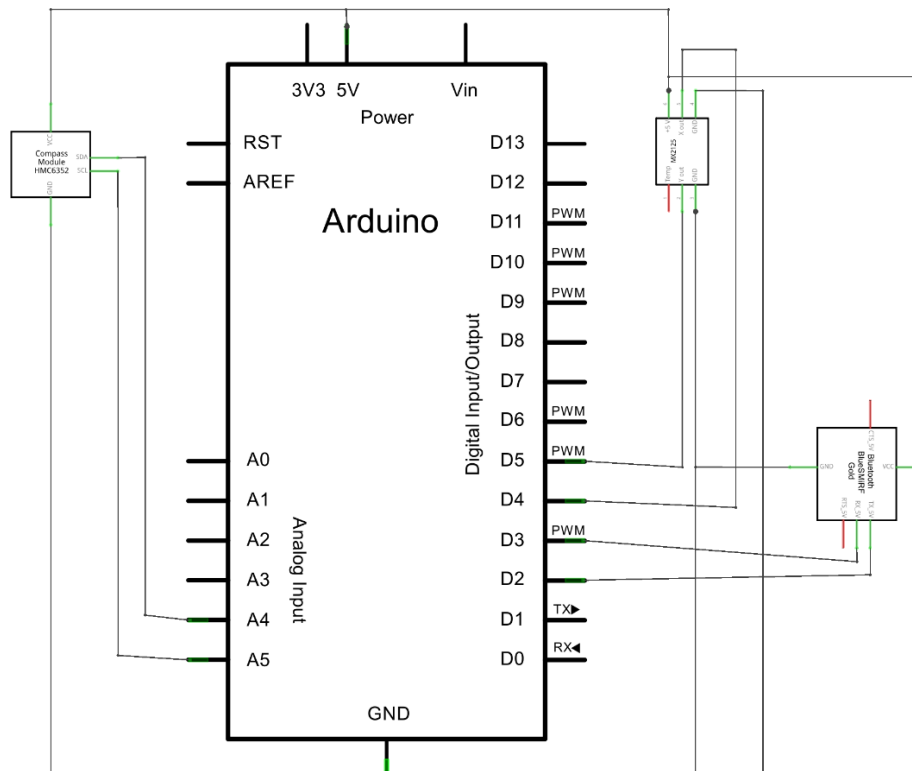


Figure 13. Input Device Assembled

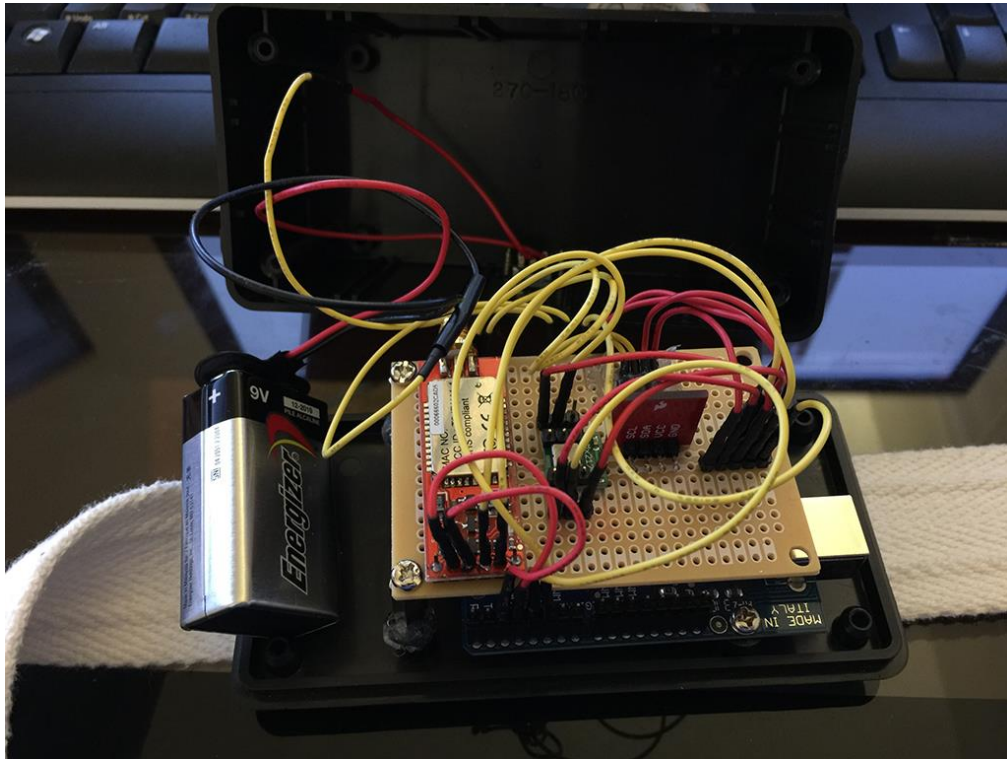


Figure 14. Input Device Assembled Angled View

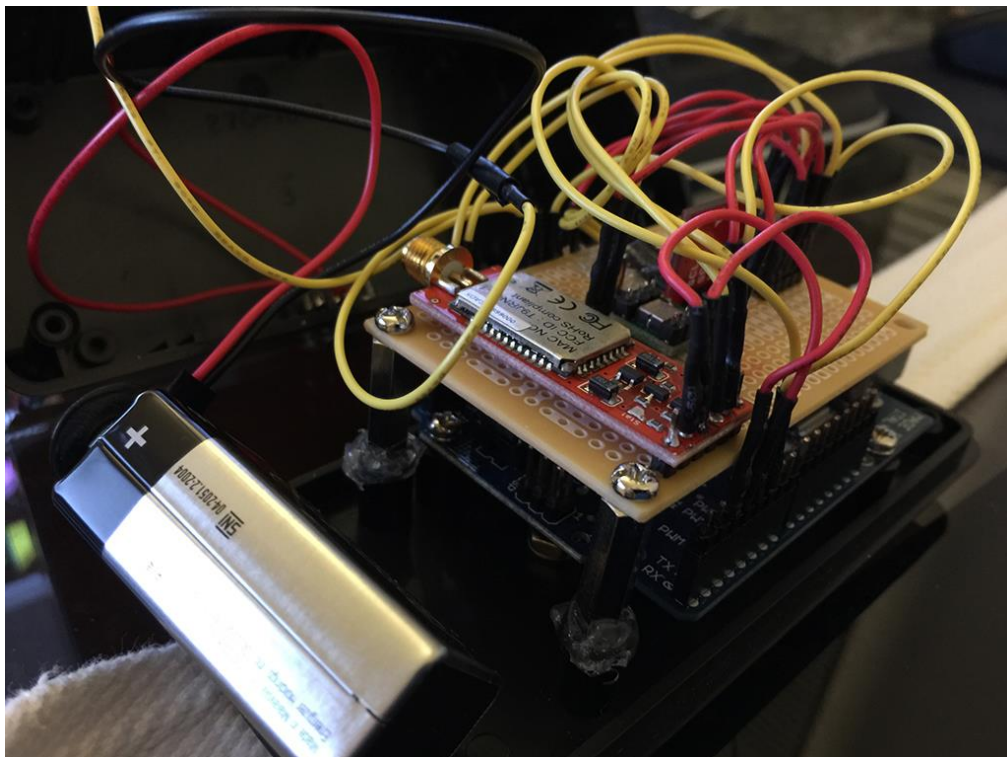


Figure 15. Input Device Attached to Belt



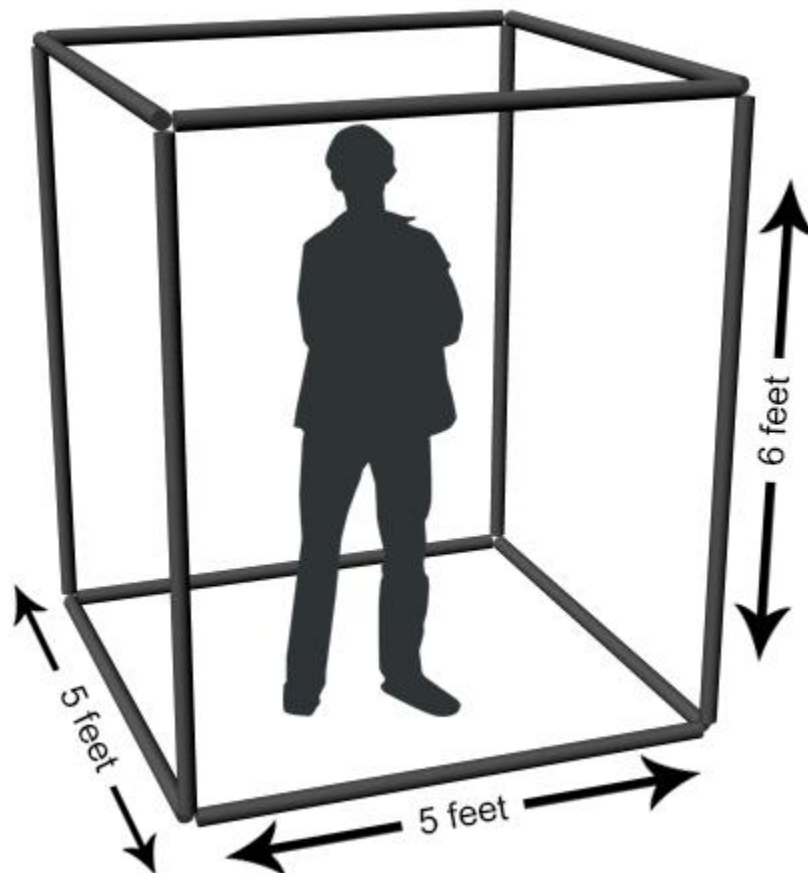
5.2. PROJECTION CUBE

To create a fully immersive gaming experience, the player needs to have a 360 degree view of the virtual environment. This is achieved by surrounding the player on four sides with a translucent material, and projecting reversed images onto that material from outside the cube. Four cameras are used in the game with their fields of view aligned to capture a seamless 360 degree view around the player.

The frame of the cube was built using $\frac{1}{2}$ " PVC pipe because it's sturdy, lightweight, and easy to assemble using pipe fittings. The cube is 5x5x6' to allow ample room for movement. Stretched across the sides of the frame are four shower curtain liners, attached to the top horizontal pipes with white duct tape, and the side vertical

pipes with several pieces of Velcro. Shower curtain liners are very inexpensive and translucent enough to clearly display projected images to the player inside. Velcro was used on the vertical pipes to allow for adjustments of the liners, ensuring a taut viewing surface which provides the clearest image. Additionally, the Velcro allows for easy entry and exit of the cube.

Figure 16. Projection Cube



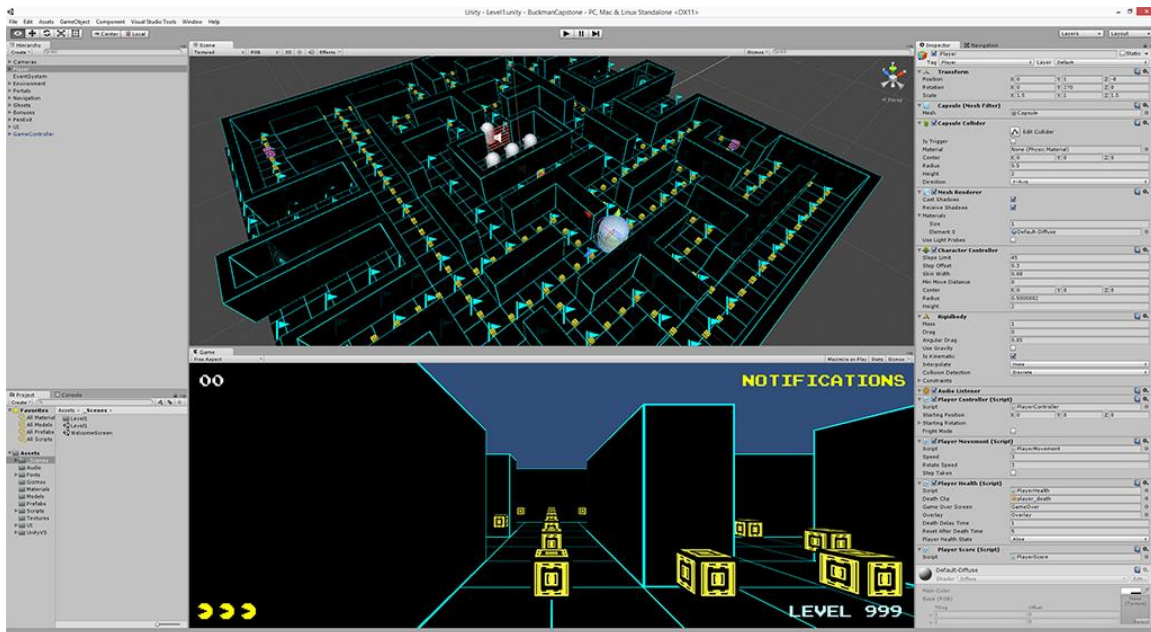
5.3. SOFTWARE

5.3.1. GAME ENGINE

As previously stated, attempting build my own game engine in a 20 week time period was a foolish endeavor. To complete the project in a reasonable amount of time I needed to switch to a pre-built game engine. After reviewing the available products, I decided to use the Unity game engine.

Unity is a free rendering engine that comes with an incredibly powerful editor that includes an extensive set of tools. It facilitates the rapid development of quality video games, providing the tools needed for building scenes, scripting, animation, user interfaces, and much more. Unity supports three programming languages: C#, JavaScript, and Boo, a .NET language similar to Python. All code for this project was written in C#.

Figure 17. Screenshot of Unity Editor



5.3.2. INTERFACING WITH THE INPUT DEVICE

The game interfaces with the input device through a wireless Bluetooth connection. Once the game is launched, the wireless connection is established through a screen in the options menu. A controller class handles the work of establishing the connection to the input device as well as sending any commands and listening for the responses to those commands. The controller class also creates and maintains instances of the `SerialListener` class, which runs on a separate thread and has the sole purpose of listening for data from the Arduino microcontroller. The reason the `SerialListener` class runs on a separate thread is that whenever a call is made to read data from the serial port, the thread is paused until a response is received or the request times out. In addition to causing stuttering of gameplay when serial reads cause delays, the game is completely frozen for any length of time that serial data is not received.

After the connection has been established, the player must calibrate the input device. Additionally, the device must be calibrated whenever a new player wears it. The calibration procedure sets the base acceleration and heading values. The player faces the forward screen during the procedure, setting that as the forward direction. Once the device has been calibrated, the microcontroller runs a continuous loop checking for significant changes in vertical acceleration. If such a change is detected, the microcontroller requests a heading reading from the compass sensor. Once this heading is received, a rotational offset is calculated based on the heading determined to be forward during the calibration procedure. This value is then transmitted via the

Bluetooth module to the game, where it's picked up by the serial listener. The value signals to the game that a step was taken, and the rotational value is applied to the player game object before applying acceleration to the object, moving the player in the direction he is facing.

5.3.3. ENVIRONMENT

The game environment is a 3D replica of the original *Pac-Man* labyrinth. It consists of a series of intersecting paths, with a pen to hold the ghosts in the center of the level and two tunnels on either side which lead to portals that transport the player and ghosts to the tunnel at the opposite side. The walls are built from a small set of models representing all possible pieces of a labyrinth. This modular approach makes modifying and building new levels very easy.

Figure 18. Labyrinth Layout

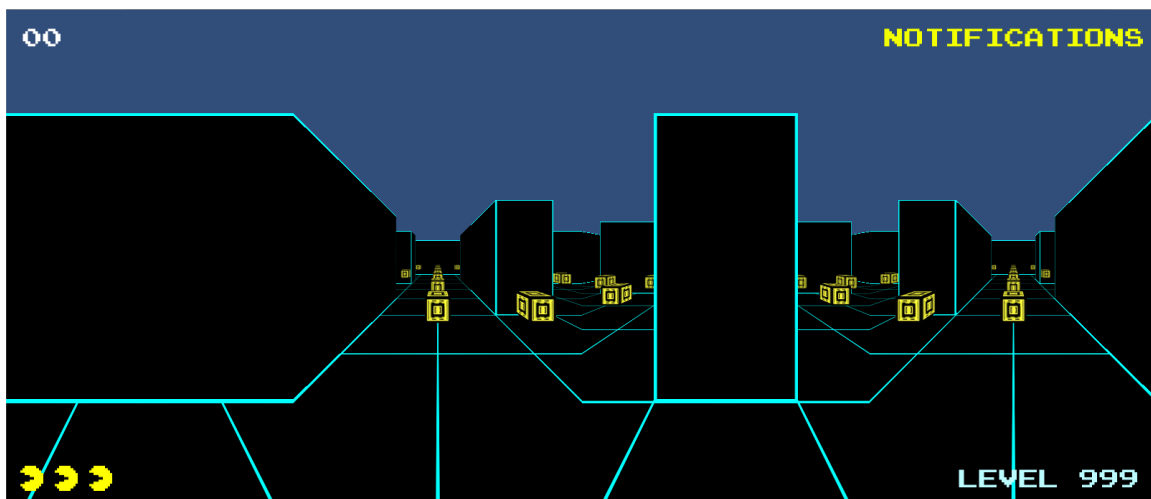


5.3.4. PLAYER

The game is played from a first person perspective, so there is no visual representation of the player. The player game object consists of a collider object used to detect collisions with other objects, a character controller and rigid body to allow for movement, and four cameras facing away from the player. The four cameras face in all directions, with the edges of their fields of view lining up to create a seamless 360 degree view of the environment.

Player movement is affected by data received from the input device. As a player walks or runs in place, the microcontroller sends information to the game regarding the player's heading. When this is received, the player game object is rotated to match the heading and acceleration is applied to the object. Gravity is constantly applied and quickly causes the player object to stop if no steps are detected. This building up and gradual lessening of acceleration provides for much smoother movement through the scene.

Figure 19. 360 Degree Camera View



5.3.5. GHOSTS

The ghosts were modeled after the enemies from the original *Pac-Man* game. They include a model of the ghost body and a separate model of the eyes. The texture applied to these models is changed depending on the current state of the ghost—red, teal, pink, or orange for the normal state, blue for the frightened state, and flashing blue and white when exiting the frightened state. Only the eyes are visible after a ghost has been eaten until it returns to the pen.

Ghost movement is dictated by a grid of waypoints, lined up with all of the intersections within the labyrinth. Figure 20 below shows this grid of waypoints, marked by the teal flags in the image. The figure also shows the baked in navigation path, which restricts the movement of the ghosts. Movement starts for the ghost by exiting the pen moving to the nearest waypoint just outside. From here, an endless series of movements from waypoint to waypoint begins. Each time a ghost reaches its destined waypoint, it gets a list of all navigable adjoining waypoints excluding the one behind it. From this list it randomly selects a new waypoint, sets it as its target destination, and continues on its path.

There are two circumstances under which a ghost will stray from the default movement logic. When ghosts change modes they immediately reverse direction, and when a ghost is in scatter mode it travels between a set of waypoints in its assigned corner which cause it to circle its home area. The speed at which a ghost moves is also

impacted by two special situations. Movement is slowed when in the frightened state and also when traveling down one of the two tunnels on the side of the board.

Figure 20. Waypoints and Navigation Path

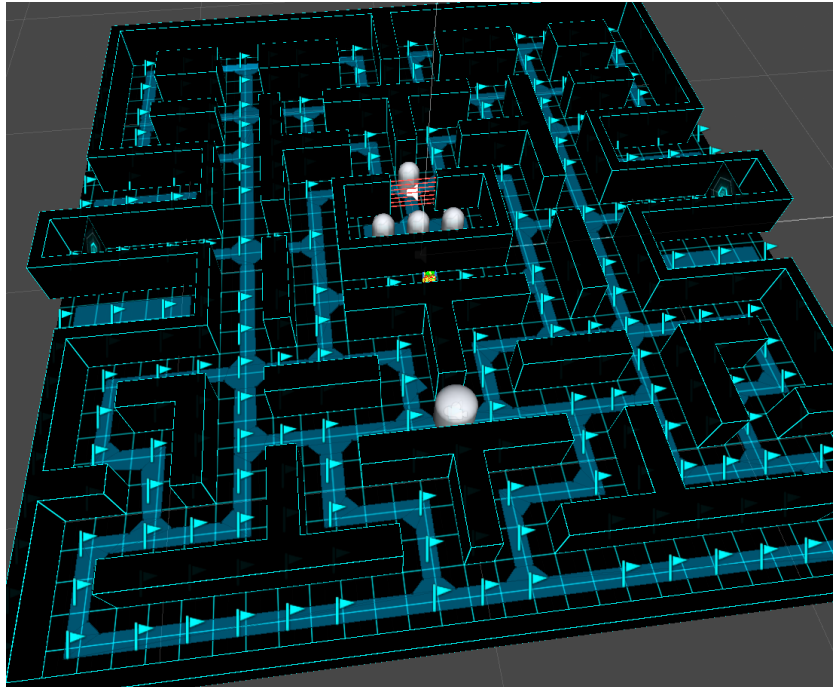
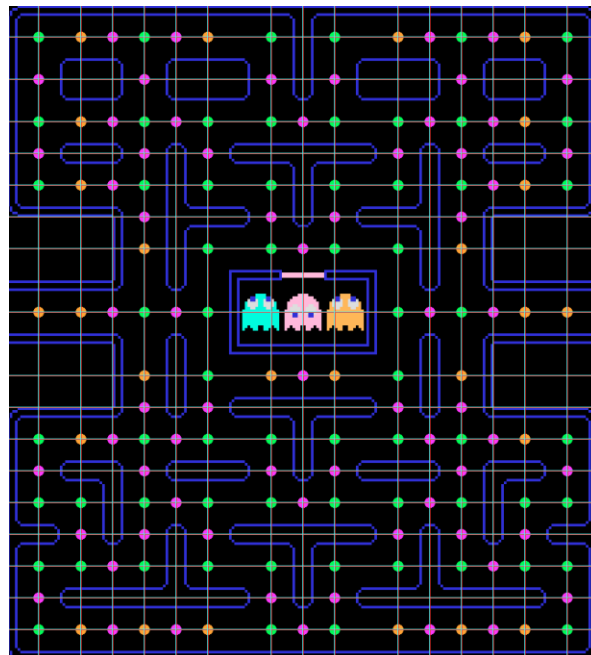


Figure 21. Waypoint Grid



5.3.6. GAMEPLAY

The object of the game is to score as many points as you can before losing all of your extra lives. Points are scored by navigating the labyrinth and eating dots. Four special dots called energizers exist on each level. In addition to being worth more points, they also send the ghosts into the frightened state. The player can eat ghosts in the frightened state for bonus points, which double each time you eat a ghost for the duration of the current frightened state. Another way to score more points is to eat the bonus fruit which appear twice per level near the ghost pen after a certain number of dots have been eaten.

If the player is caught by a ghost, he loses one of his extra lives. The player begins the game with two extra lives (for a total of three lives). If the player loses all extra lives then the game is over and he must start a new game. The player is awarded a single additional extra life after scoring 10,000 points.

Difficulty is increased each level by adjusting various gameplay settings. Properties such as player speed, ghost speed, how quickly ghosts are released from the pen, and the length of time ghosts remain in frightened mode are altered to make the gameplay more challenging. Some of the difficulty variables from the original *Pac-Man* game were left out as they made this form of the game too difficult. In the original game, ghosts periodically enter a mode where they directly target the player. After a few rounds the ghosts remain in this chase mode. Additionally, after a certain number of dots have been eaten, the ghost, Blinky, increases his speed. Given the fact that

more effort is required to navigate the board in the physical version of the game and the user has no top down view of the board to provide awareness of ghost locations, these new challenges provide an offset to the old ones and balance out the difficulty.

Figure 22. Level Specifications

Level	Bonus Symbol	Bonus Points	Pac-Man Speed	Ghost Speed	Ghost Tunnel Speed	Fright. Pac-Man Speed	Fright Ghost Speed	Fright. Time (in sec.)	# of Flashes
1	Cherries	100	80%	75%	40%	90%	50%	6	5
2	Strawberry	300	90%	85%	45%	95%	55%	5	5
3	Peach	500	90%	85%	45%	95%	55%	4	5
4	Peach	500	90%	85%	45%	95%	55%	3	5
5	Apple	700	100%	95%	50%	100%	60%	2	5
6	Apple	700	100%	95%	50%	100%	60%	5	5
7	Grapes	1000	100%	95%	50%	100%	60%	2	5
8	Grapes	1000	100%	95%	50%	100%	60%	2	5
9	Galaxian	2000	100%	95%	50%	100%	60%	1	3
10	Galaxian	2000	100%	95%	50%	100%	60%	5	5
11	Bell	3000	100%	95%	50%	100%	60%	2	5
12	Bell	3000	100%	95%	50%	100%	60%	1	3
13	Key	5000	100%	95%	50%	100%	60%	1	3
14	Key	5000	100%	95%	50%	100%	60%	3	5
15	Key	5000	100%	95%	50%	100%	60%	1	3
16	Key	5000	100%	95%	50%	100%	60%	1	3
17	Key	5000	100%	95%	50%	–	–	–	–
18	Key	5000	100%	95%	50%	100%	60%	1	3
19	Key	5000	100%	95%	50%	–	–	–	–
20	Key	5000	100%	95%	50%	–	–	–	–
21+	Key	5000	90%	95%	50%	–	–	–	–

6. PRODUCT COMPARISON

When the proposal was first written, the Nintendo Wii was the only commercially available product which offered an experience in any way similar to this project. Since then, several other products have been released offering physical interactive gaming experiences. This section will review some of these more popular products and compare the experience they provide with that provided by this project.

6.1. ACCELEROMETER TRACKING

Accelerometer tracking makes use of sensors to detect player tilt, rotation, and movement in various directions. This data is then used to affect movement of player avatars or other aspects of gameplay. The primary product to make use of this form of

tracking is the Nintendo Wii system through the Nintendo Wii Remote. Some fitness games developed for the Nintendo Wii use accelerometer readings from the Wii Remote to detect player steps, treating it like a pedometer. This behavior is similar to that of the input device used for this project, but is lacking in a couple ways.

First, the Wii Remote is meant to be placed in the player's pocket. If the player is wearing clothing without pockets, the remote is instead held in the player's hand. This forces the game to rely on activity detected when the player moves his arms, and may not accurately reflect actual walking or running. This can lead to less accuracy and a less immersive experience.

A second shortcoming stems from the fact that these interactions are intended to happen with only one display. This means that an additional input device must be used to control player movement outside of simple forward momentum. This also

Figure 23. Nintendo Wii Remote



means that these interactive games are incapable of offering a fully immersive gaming experience.

6.2. CAMERA TRACKING

One of the most popular methods of detecting player movement is through the use of cameras. These cameras continuously capture images of the player and their environment which are used by the system to interpret movement. Two available products using this technology are Xbox Kinect and PlayStation Camera. Both of these systems make use of image capture to detection motion, but do so in slightly different ways.

The Kinect 2.0 device used with Microsoft's Xbox One system consists of a single 1080p RGB camera used for capturing high definition images. The Kinect's motion detection is augmented by the use of an infrared blaster which sends out particles of light that bounce off the player and their environment. The amount of time taken for these particles of light to return to the camera is used to determine distance, very similar to how sonar imaging works.

Sony's PlayStation Camera device does not have an IR blaster, but does make use of a second camera to improve the accuracy of its motion detection. The two cameras work together to triangulate the player's position. This dual camera setup improves accuracy and allows Sony to ditch the wand controllers required by PlayStation Camera's predecessor, PlayStation Eye.

Both camera tracking devices offer application control and interaction through gesturing, something this project lacks but could make great use of. Gesture control could further remove connections to the world outside the projection cube, offering an

even more immersive experience. However, like the Nintendo Wii, these products are intended for use with a single display, again depriving the user of an experience and environment that fully surrounds them.

Figure 24. PlayStation Camera and Xbox Kinect



6.3. VIRTUAL REALITY HEADSETS

One immersive experience which has gained a lot of traction as of late is that provided through the use of virtual reality headsets. These devices are worn over the user's eyes, completely covering them to block out all other views and completely immerse the user in the virtual world. Screens inside the device display a view of the 3D virtual environment which is adjusted based on the user's movement. While Sony is working on their device, Morpheus, and Google has released Google Cardboard which uses an Android smartphone and cardboard goggles built from a template to create a VR headset, the most prominent VR headset today is the Oculus Rift.

In addition to the goggles housing the display screens, the Oculus Rift contains a control box attached to the headset with a 6 foot cable, as well as several sensors used

to read user movement. These sensors include a gyroscope, accelerometer, and magnetometer, two of which are used in the input device for this project. While the original device only allowed for the tracking of player orientation, newer iterations are including infrared LEDs on the device which are monitored by an external camera. This provides some positional tracking and allows players to lean in for closer views or tilt their heads to look around corners.

Virtual reality headsets definitely have the edge when it comes to full immersion into a virtual environment. The devices are able to completely block out the external world in a way that this project is incapable. What is lacking, though, is the health benefit provided by this project. While the VR headsets could likely be used to track a user's movement in a way that facilitates travel through a virtual world, it's accompanied by a danger to the user. Having no view of the real world around you places the user at risk of walking into objects, tripping, or becoming disoriented and falling down. The cable permanently attached to the headset adds an additional risk of injury. These devices have also been known to cause headaches and motion sickness in some users. So, while it's a more immersive experience, VR headsets do not seem well suited for vigorous physical activity.

That being said, use of a VR headset with additional equipment, namely the Virtuix Omni, could alleviate some of the issues previously stated. The Virtuix Omni is a treadmill system with a support harness attached to and support ring surrounding the user. The user is able to walk or run on the platform, which moves under their feet simulating real world movement. While this doesn't help with the issues of

disorientation and motion sickness, it is a great solution to the previously stated safety concerns and is one of, if not the, best way to handle movement through an immersive virtual environment.

Figure 25. Oculus Development Kit



Figure 26. Oculus Rift Inside View



Figure 27. Virtuix Omni



Figure 28. Virtuix Omni in Use



7. CONCLUSIONS

7.1. LESSONS LEARNED

The most valuable lesson learned from this project is to set realistic goals. A Capstone project is intended to show your skill and what you learned throughout the course of your graduate studies. It is not, however, a time to learn completely new topics and attempt to reinvent the wheel. Because I decided to tackle something new and go way above and beyond the requirements of a Capstone project, an incredible amount of additional work was imposed and the project strayed greatly from the established work timeline. As a result, a large amount of completed work eventually had to be scrapped and redone in a much smaller timeframe. Additionally, delays resulting from that original derailment imposed extra hurdles due to facility and other changes occurring as time went by and the project stagnated. For these reasons it is imperative to set realistic goals and deadlines at the start of a project, or you will be very likely to fail.

A second lesson learned is that the translation of an application from a traditional format to an immersive one is accompanied by a whole host of obstacles one might not immediately think of. Particularly impacted is the display of textual or other two-dimensional data, like that shown in a heads up display (HUD). Moving the HUD on the screen as the user rotates presented issues because the heading received constantly varied slightly, resulting in a jittery display. Displaying the HUD stretched across all screens spread the information out too much, and showing it on all screens was

repetitive and cluttered up the view. The decision was made to only display the HUD on the screen considered forward, that is the one the player faces at the start of the game. This served our application well since the only time a view of the HUD is vital to the player is at the start of the level. Displays with more pertinent information would not be best incorporated using this method, and an alternative solution would likely need to be implemented.

Another impact of the translation to an immersive format was the loss of a top down view of the environment. In the original *Pac-Man* game, the player had a view of the entire level and could see when ghosts were around an upcoming corner. This is not the case when the player is placed inside of the environment. The loss of that all-encompassing view increased the difficulty of the game, but also made it more exciting. Users must now attempt to listen for the locations of ghosts and need quicker reflexes to alter course if a ghost crosses their path. Nevertheless, the difficulty of the game was increased. To counter this, some features of the original game were left out, such as periodic increases in ghost speeds and “chase mode”, where ghosts would relentlessly target the player.

The change in input device from keyboard to human movement also instilled the need to make adjustments to the game. The new method of input is far more physically demanding, which serves the goal of promoting health benefits, but could also exhaust the player before the first level is complete if not handled correctly. For this reason, the speed of the player’s movement through the environment had to be tweaked to create

an experience that provided aerobic value, but was also enjoyable and sustainable to play.

A third more minor, yet still valuable, lesson learned was to always thoroughly research and test your hardware components. The compass module used in the input device had been well-tested individually, as well as with other components. However, when the entire device was assembled an issue was discovered. Lacking an accelerometer to account for tilt, the compass module must be kept level in order to receive accurate readings. The sensor had been soldered to the circuit board in a way that positioned it at a 90 degree angle when worn by the user, resulting in inaccurate and unusable data. As a result, the soldering had to be undone and the sensor reattached in a way that positions it flat when worn by a user.

7.2. FUTURE WORK

One strong potential addition to the project would be integration with a handheld device. As a solution to the previously stated problem of displaying information, a tablet or smartphone could be integrated with the system to serve as the main point of information for the player. Real-time scoring and notification data could be continuously streamed and displayed on the device. A most valuable feature would be a mini-map or radar system, providing the player with real-time information on the whereabouts of enemies and valuable targets.

Improvements could also be made to the input device. Using a 9 volt battery to power the device works for short-term demonstration, but is inefficient and expensive

for long-term use. Additionally, while the detection of footsteps is pretty accurate, improvements could be made to eliminate false readings, missed readings, and even to incorporate jumping into the available interactions. An upgraded compass sensor with tilt compensation could also prove very useful in improving the player's experience and making for more reliable and accurate movements.

Another area to explore would be the use of this setup in applications other than gaming. The system could be used to participate in virtual tours of real places—like Google Earth but more immersive. One could travel the world without ever having to set foot on a plane. Aside from real locations, artificial environments could be created as well to explore and serve as a motivation to exercise or even a space of tranquility and relaxation.

A final example of future work is to incorporate the software application with smartphones or other mobile devices. Most smartphones today include all of the sensors put into the input device for this project. Altering the application to work with these devices would allow a player to simply use their own equipment when entering the cube to play the game. This could simplify and expedite entering a game, and also allow for the creation of applications which connect and accept input from multiple players at the same time.

7.3. CONCLUSION

While strides are continuously being made in the fields of immersive gaming and tangible interfaces, there's still an incredible amount of potential within these areas.

The video game industry is a huge and ever-expanding one with much untapped potential, but also one with some negative impacts on society. Increased use of video games is a continuous contributor to the growing epidemic of obesity in the United States and other countries. Tangible interfaces offer a way to not only help combat the epidemic, but also contribute to the growth and profitability of the gaming industry. Incorporating natural, physical interactions with immersive environments helps draw the player into the virtual world, blurring the boundaries between realms and creating a richer experience.

8. BIBLIOGRAPHY

1. Baranowski, T. (2013, June 11). Games and Childhood Obesity. In *Games for Health Journal*. Retrieved November 26, 2014, from <http://online.liebertpub.com/doi/full/10.1089/g4h.2013.1502>
2. Beckhaus, S., Blom, K. J., & Haringer, M. (2005). *Intuitive, Hands-free Travel Interfaces for Virtual Environments*. Retrieved May 8, 2008, from University of Hamburg Web site: <http://imve.informatik.uni-hamburg.de/files/19-IntuitiveTravel-VR2005-BeckhausBlomHaringer.pdf>
3. Dyer, O. (2007, March 15). Can Video Games Zap Childhood — and Adult — Obesity? In *National Review of Medicine*. Retrieved May 8, 2008, from http://www.nationalreviewofmedicine.com/issue/2007/03_15/4_advances_medicine_5.html
4. Goldfield, G., Cameron, J., Chaput, J. (2014, March 11). Is Exergaming a Viable Tool in the Fight against Childhood Obesity? In *Journal of Obesity*. Retrieved November 26, 2014, from <http://downloads.hindawi.com/journals/job/2014/304521.pdf>
5. Gudmundsen, J. (2007, October 8). Video Game Tackles Childhood Obesity. In *ABC News*. Retrieved May 8, 2008, from <http://abcnews.go.com/Technology/Story?id=3695822&page=1>
6. Johnson, Bernadette. (2014, March 7). *How the Oculus Rift Works*. Retrieved November 22, 2014, from HowStuffWorks.com: <http://electronics.howstuffworks.com/oculus-rift.htm>

7. Maddison, R. (2011, May 11). Effects of active video games on body composition: a randomized controlled trial. In *The American Journal of Clinical Nutrition*. Retrieved November 26, 2014, from <http://ajcn.nutrition.org/content/94/1/156.short>
8. Marshall, P., Rogers, Y., & Hornecker, E. (2007). *Are Tangible Interfaces Really Any Better Than Other Kinds of Interfaces?* Retrieved May 8, 2008, from Open University, Pervasive Interaction Lab Web site: <http://www.cl.cam.ac.uk/conference/tangibleinterfaces/TUIworkshop-Marshall.pdf>
9. Mazalek, A., & Nitsche, M. (2007). *Tangible Interfaces for Real-time 3D Virtual Environments*. Retrieved May 8, 2008, from Georgia Institute of Technology Web site: http://synlab.gatech.edu/data/papers/mazalek_ace2007_tui3d.pdf
10. Ogden C., Carroll M., Kit B., Flegal K. (2014). Prevalence of Childhood and Adult Obesity in the United States, 2011-2012. In *The Journal of the American Medical Association*. Retrieved November 30, 2014, from <http://jama.jamanetwork.com/article.aspx?articleID=1832542>
11. Thompson, D. (2014, July 16). What Serious Video Games Can Offer Child Obesity Prevention. In *JMIR Serious Games 2014*. Retrieved November 26, 2014, from <http://doi.org/10.2196/games.3480>
12. Torgan, C. (2002, June). Childhood Obesity on the Rise. In *The NIH Word on Health*. Retrieved May 8, 2008, from <http://www.nih.gov/news/WordonHealth/jun2002/childhoodobesity.htm>
13. Tse, E., Greenberg, S., Shen, C., & Forlines, C. (2006, May 7). Multimodal Multiplayer Tabletop Gaming. In *Third International Workshop on Pervasive Gaming Applications*. Retrieved May 8, 2008, from <http://www.merl.com/reports/docs/TR2006-009.pdf>
14. Warner, J. (2004, July 2). Video Games, TV Double Childhood Obesity Risk. In *WebMD Medical News*. Retrieved May 8, 2008, from <http://children.webmd.com/news/20040702/video-games-tv-double-childhood-obesity-risk>
15. Wisneski, C., Orbanes, J., & Ishii, H. (1998, April 18). *PingPongPlus: Augmentation and Transformation of Athletic Interpersonal Interaction*. Retrieved May 8, 2008, from MIT Media Laboratory Web site: http://tangible.media.mit.edu/content/papers/pdf/PingPongPlus_CHI98.pdf