Senior Design Final Report

Eben Blaisdell[+], Dale Hartman[+], Sean McDonough[+], and PJ Onusconich[+]

[+]these authors contributed equally to this work

1 Overview

1.1 Motivation

Our team was tasked by Professor Erin Jablonski and the Lewisburg Children's Museum to create an application that virtually displays projectile motion. This projectile motion simulation needs to be fun and interesting to kids between the ages of 4 and 12 while also facilitating their understanding of projectile motion concepts. While it is not constrained to the real world, the museum must be able to build a physical model exhibiting the main features of the simulation.

1.2 Solution

Our team's solution was to design a game centered around concepts of projectile motion. The core features of this game included launching a ball from a turret in order to knock blocks off platforms. By adding different block structures and challenges, we would create a game that would excite the children interacting with it while also being simple enough in nature to be replicated by the museum.

2 Project Description

2.1 Major Features

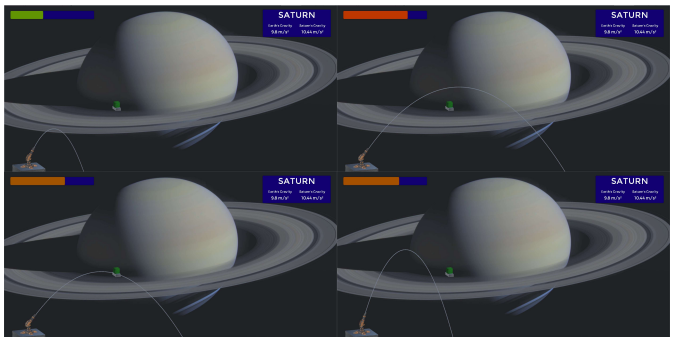

Figure 1. Varying angles and velocities the turret can launch projectiles with.

- Users are able adjust turret angle and power to launch projectiles at various angles and velocities.

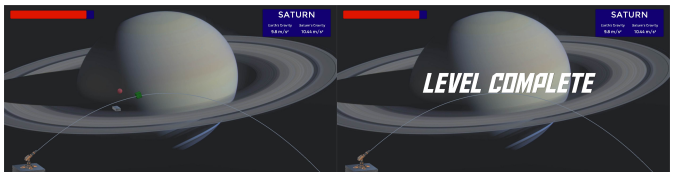

Figure 2. Knocking over green blocks advances the level.

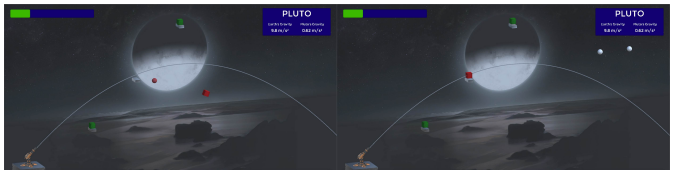

Figure 3. Knocking over red blocks resets the level.

- Users can utilize projectiles to knock over blocks on screen. Knocking over all green blocks will advance the level. Knocking over a red block resets the current level.

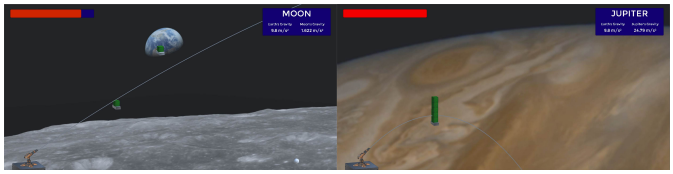

Figure 4. Example of different locations for levels: the Moon and Jupiter.

- Users can complete levels that are set on different celestial bodies in the Solar System. The celestial body that the level is set on effects the gravitational force acting on the ball. Levels are designed with different block structures and obstructions and difficulty increases as the user progresses through the game.

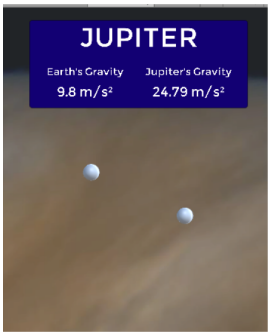

Figure 5. Ball on left with Earth's gravity vs. ball on right with Jupiter's gravity

- Users can compare the gravity of the current celestial body as compared to Earth through two dropping balls at the start of the level. The ball on the left displays Earth's gravity and the ball on the right displays the celestial body's gravity. UI text displays the gravitational value.

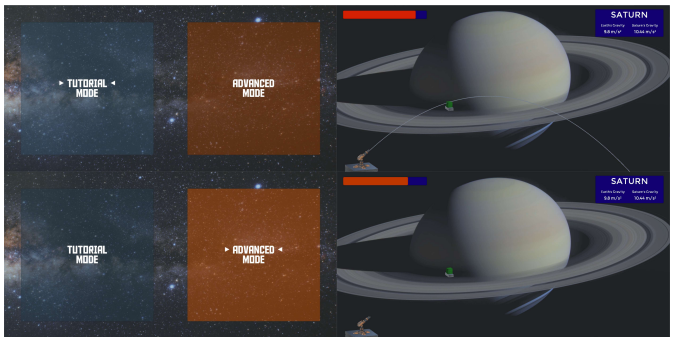

Figure 6. Comparison of first level of tutorial mode vs. advanced mode.

- Users can select between tutorial and advanced modes befores starting the game. Tutorial mode is meant for new users providing in-game instructions and a predicted projectile path. Advanced mode is meant for experienced users and removes instructions and the predicted path.

2.2 Demo Video

A demo video for our project can be found at: https://www.youtube.com/watch?v=N43o9R1F-V0

3 System Design

3.1 Overview

Figure 7. The Hardware Setup

The overall system consists of a desktop PC system provided by the museum, with a monitor and a pair of speakers, and a custom-built set of hardware controls based around an Arduino. The Unity game runs on the PC, and users interact with the system only through use of the custom controls.

3.2 Software

3.2.1 Software Choice

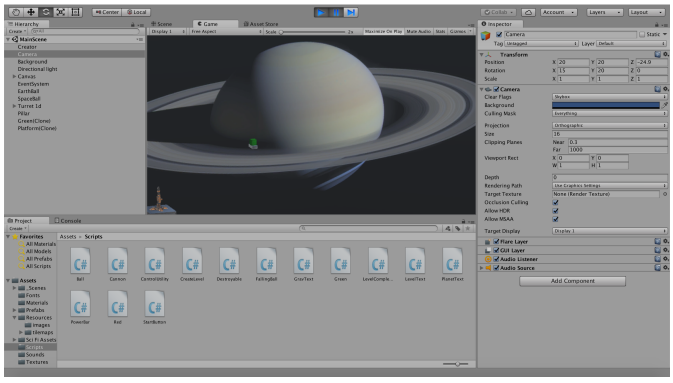

Figure 8. Screen capture of the Unity editor

We utilized the unity game engine Unity as the software for developing our application. Since Unity is primarily used for the development of video games, our gamified solution to teach projectile motion to kids made this an appropriate choice. The back end is coded in C# scripts and heavily utilizes Unity's built in physics engine to handle the launching of projectiles and collisions.

3.2.2 Control Scheme

Controls in the game are based on mouse and keyboard inputs. Mouse movements in the x and y directions alter angle and power respectively and left clicking launches a projectile. Pressing the S key resets the game and the right arrow key advances the game to the next level, the latter of which was implemented for debugging purposes. Creating a control system in this fashion interfaced the software with the hardware's mouse and keyboard emulation and also allowed for the software to be tested without the need for a functional controller.

3.2.3 Level Editing

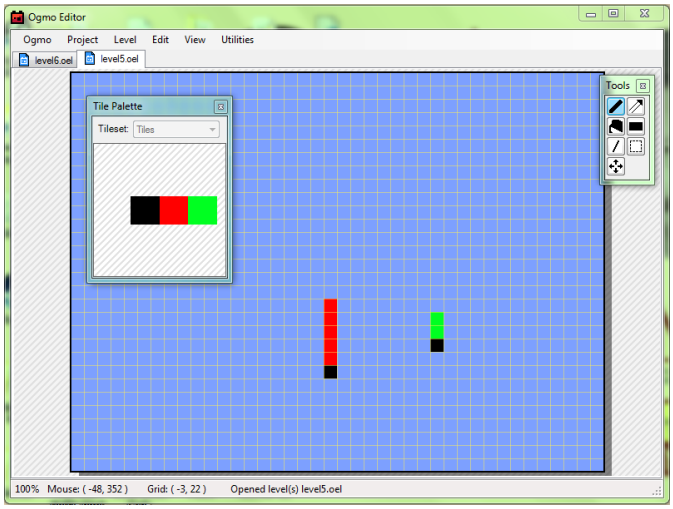

Figure 9. Screen capture of the Ogmo editor

Level design was done through the use of the tile editing software, Ogmo[1]. In the Ogmo editor, a specified tile can be selected and added to the scene. Ogmo oel files are similar to XML in nature and contains a csv representing the tile grid under one of its tags. We extracted this csv and inserted into a new csv file representing the level with the first two lines of the new file indicating the celestial body and that celestial body's gravity respectively. Using a C# script, the csv is parsed, setting the background to the appropriate image, the gravity to the appropriate value, and building the level based on the csv extracted from the Ogmo oel file.

3.3 Hardware

3.3.1 Hardware Overview

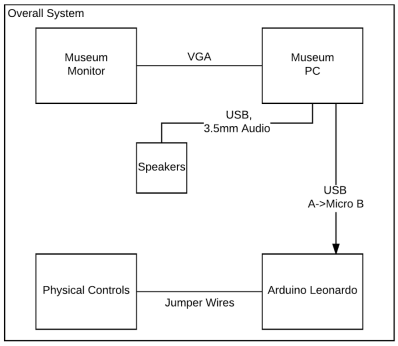

Figure 10. System Design Diagram

The hardware system is designed around a small form factor desktop PC provided by the museum, which runs the software component. The PC is hooked up to a monitor via VGA, and to a pair of speakers via USB and 3.5mm audio connectors, all of which are also provided by the museum. For user interaction, the PC is connected via a USB A to Micro B cable to a custom arcade controller built around an Arduino Leonardo.

[1] http://www.ogmoeditor.com/

3.3.2 Hardware Controller

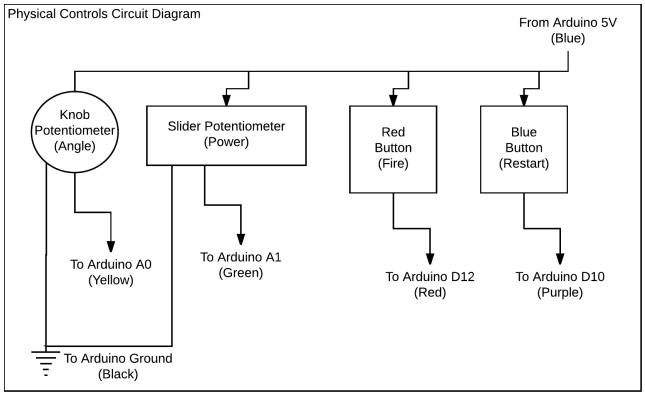

Figure 11. Physical Controls Circuit Diagram

Through the controller, users are able to interact with the software by changing the angle of the cannon (yellow knob potentiometer), changing the power of the cannon (green slider potentiometer), firing the cannon (red button), and resetting the game (blue button). The Arduino and control elements are housed in a (fairly sturdy) cardboard box, which can be opened to allow access to the wiring inside. For ease of maintenance, all wires within the box follow a color scheme, using blue for +5V, black for ground, and various colors roughly corresponding to each control element to carry that element's signals to the Arduino.

3.3.3 Arduino Code

In terms of the program run by the Arduino, several Arduino libraries are used to create the desired functionality. First and foremost, the Arduino Keyboard and Mouse library[2] is used to simulate control inputs as keypresses, mouse movements, and mouse clicks to send to the computer as input for the software. Secondly, the Arduino Millis library[3] is used to implement timesharing between each control element. During every forty milliseconds, the Arduino listens for input from each button and potentiometer in turn, and acts accordingly.

4 Team Processes

4.1 Overview

4.1.1 Team Roles

- Sean - Lead Hardware Engineer
- Eben - Lead Software Engineer
- Dale - Scrum Master
- PJ - Product Owner

4.1.2 Version Control/Scrum

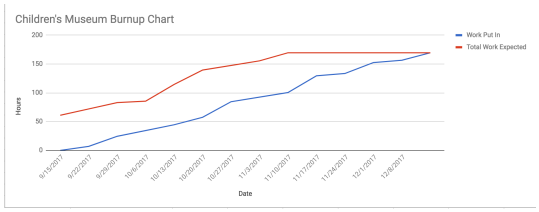

Figure 12. Our team's final burn up chart

Our team has been using Git for version control and Trello to keep up with agile/scrum methodologies.

[2] https://www.arduino.cc/reference/en/language/functions/usb/keyboard/
[3] https://www.arduino.cc/reference/en/language/functions/time/millis/

4.2 Software Development

4.2.1 Minimum Viable Product

Our first goal was to make some kind of interactive environment in which to test out the physics system. At this stage, the game was a few levels consisting of a few green blocks each. The cannon had power and angle controls and fired cannonballs, but there were no limitations as to how many balls could be fired, or how quickly. Every object in the game was using placeholder assets, and no audio or music was implemented. This stage of development allowed us to show our concept to our client, and receive the greenlight to develop it further.

4.2.2 First Demo Project

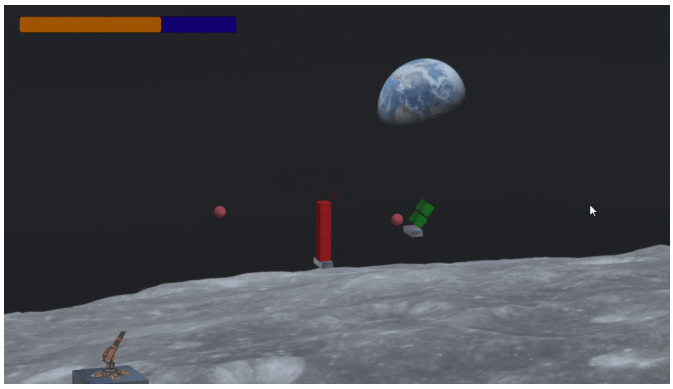

Figure 13. Game shortly before the first demo, red blocks and power bar

In the time leading up to our first user testing opportunity, we focused on making the game presentable to our target audience. The first goal was to provide the UI elements necessary to play the game successfully. A power bar was implemented that allowed players to know the exact angle and power the cannon was set to. Our assets and textures were finalized, giving the game a more polished look. Lastly we implemented more levels using red blocks, adding an element of challenge to the game.

4.2.3 Second Demo Project

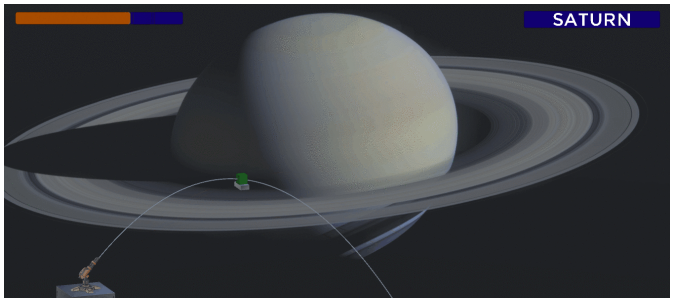

Figure 14. Game shortly before the second demo, predictive path and planets

Before our second demo, we further polished the game to something that felt closer to a final product. Music was added, and sound effects were put in for every type of collision and game cue. The predictive path was programmed to lessen the amount of trial-and-error necessary to complete the more difficult levels. To expand the content of the game, we introduced different planets into the game. Each level was to take place on a particular planet, and each planet has a different value of gravity to take into account. The game's background was changed from level to level, and labels were added to the UI to name the planet.

4.2.4 Final Project

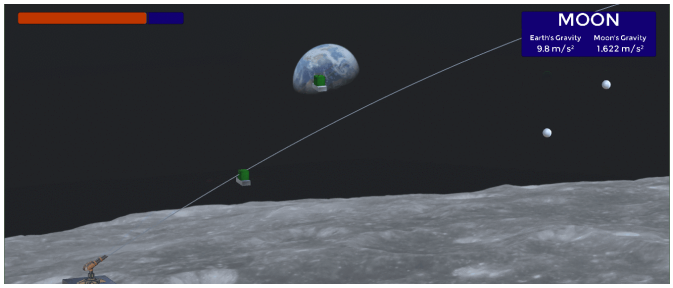

Figure 15. Gravity comparison and visual demonstration.

We received a lot of feedback from our second user demo, and our final development stretch focused on addressing these issues. The predictive path was necessary for the younger children, but made the game too easy for the younger ones. Thus, we added a main menu to the game where you can select between two difficulty modes: a tutorial mode with predictive path and more explanation of game elements, and an experienced mode with no path and no repetitive tutorialization.

A full game restart functionality was added that takes the game back to the main menu. This allows the game to set in a museum space and be played by many players of different skill levels in quick succession. Level completion transitions were added to make success more clear to the player.

The physics elements of the game were also finalized. UI elements were added that allows the children to compare the gravity of the planet they are playing on to the gravity of earth. Two falling objects at the beginning of every level visualize this difference.

4.3 Hardware Development

4.3.1 Proof of Concept

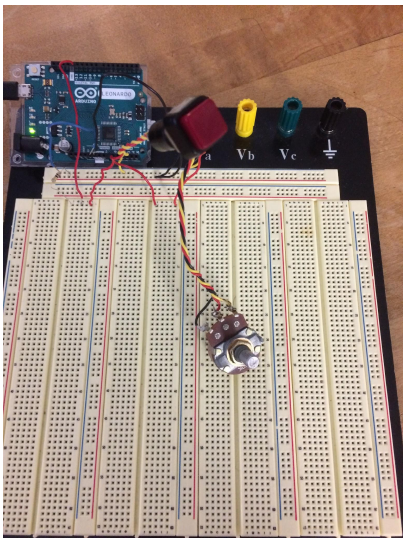

Figure 16. Hardware Proof Of Concept

The first step in the construction of the hardware controls consisted of a simple proof of concept design. The Arduino Leonardo was chosen as the heart of the controls, due to its native USB communication capabilities, which allow for the use of the Arduino Keyboard and Mouse library. This approach was chosen over interfacing directly between Unity and the hardware controls because it enabled both the hardware and software elements of the project to progress independently. While hardware controls were developed to simulate keyboard and mouse actions, the software game could be developed freely using mice and keyboards to test input.

4.3.2 Alpha

Figure 17. Alpha Controls

Moving into Alpha, the first set of usable controls consisted of the Arduino connected to a knob, slider, and a single button, all built inside a (somewhat flimsy) cardboard box. While the Arduino, knob, and button were final components, the slider was a temporary stand-in for the final slider, which was not yet available. While these controls were functional, they were not sturdy enough to last more than five minutes in a demo with children. After being dropped and broken, the alpha controls were not repaired, and the feedback and insight gained from the demo was used to construct the next iteration.

4.3.3 Beta

Figure 18. Beta Controls

Applying the lessons learned from the failure of the alpha controls, the beta controls were constructed in a significantly sturdier cardboard box. Additionally, more care was taken in designing and constructing the controls for strength and aesthetics. Internally, the wiring was custom cut to length, instead of using pre-cut wires, and wire ends were secured more reliably. In terms of the controls themselves, the button and knob from the alpha controls were added to a new button and slider, in order to allow for the full set of actions (angle, power, fire, restart). Also, the Beta controls were successfully used in the second demo without being damaged.

4.3.4 Final

Figure 19. Final Controls

While the final set of controls was intended to graduate from the cardboard housing into something stronger and better looking, such as laser cut wood or acrylic, technical issues with the laser cutter prevented that goal from being achieved. Because of this, the beta housing became the final controls. Nevertheless, some improvements were still made; handles for the knob and slider were 3-D printed with directional arrows showing how each control is intended to operate. Since the intended final version of the controls could not be created in time, the documentation for the controls is designed to make picking up where progress was left off as easy as possible. With this, the museum, or any other group they pass the project to, should have very little difficulty in creating an improved set of final controls.

4.4 Process Reflection

4.4.1 What Worked

- Unity - Unity provided us with a simple way of implementing complex physical interactions. Its built in physics engine allowed us to bypass the hassle of coding physics from scratch which enabled us to complete the core of our game in a quick manner.

- Team Roles - We assigned roles at the beginning of the semester based on individual skills. Each member of the team took to their role with little issue. Even though we lacked the recommended 'Enforcer' role, each of us policed ourselves, and we rarely ran into issues where parts of the project ran behind schedule.

- Custom Controller - This was a large investment of time and resources that fully payed off. The inclusion of physical controls was something that our client pushed us to do, avoiding the touch-screen interface that was our first idea. The tactile controls went over well with the kids in every demo we performed.

- Demos - The children seemed to be engaged with the game each time we brought it to them to demo. Each demo gave us a long list of meaningful feedback we could sort through at our next team meeting. The enthusiasm that the kids showed also boosted the morale of our group, boosting our confidence in the work we were accomplishing.

4.4.2 What Didn't Work

- Relating to a child's perspective - Our initial design for the game was based on what we believed children in our target audience would be able to do and understand. These assumptions often proved inaccurate, and through our demos we observed behavior from the players that went completely against our expectations.

- Solidifying requirements - Our client changed her mind numerous times on some of core requirements of the game. There were some elements, like the inclusion of physics formulas, that repeatedly transitioned from high-priority to low-priority. This led to wasted time researching and designing these systems, some of which never made it into the final product.

- Git - While git is a useful VCS for many projects, Unity is beyond its capabilities. The number of binary files used to keep track of unity scenes and assets led to unavoidable merge conflicts, where the only solution was to throw out the work that one of our team members had done. Redoing work we had already accomplished led to a notable amount of time lost throughout this semester.

5 Conclusion

5.1 Potential Improvements

5.1.1 Additional Obstacles and Challenges

Currently, there are limited ways we can place and structure blocks on a level. Given more time, we would add more obstacles and challenges to make each level provide a unique experience to the user. Potential obstacles could include a tunnel that players would have to guide their projectile through or angled platforms that players would have to ricochet their projectile off to knock down green blocks. Additionally, we could add challenges that will reward the player for clearing a particular level with a single projectile or knocking down multiple tower structures at once. Both of these features would allow for more total levels while also helping levels better differentiate themselves from each other.

5.1.2 Clearer Concept Communication

Our largest time sink over the course of this project was the effort we put into designing the physics learning elements the game would include. We received no specific directions from our client in this particular area, just that the game should do something to develop the children's understanding of physics. Our lack of experience in teaching these concepts, coupled with our client's inconsistent feedback on what level of physics learning we should include, has resulted in a final design that doesn't fully communicate what it needs to. Given more time, we would redesign the learning aspect of the project, perhaps adding voiced lessons or more advanced concepts for the older audience.

5.2 Wrapping Up

Overall, we are very satisfied with the project we managed to create. In the span of one semester, we designed and implemented a minimally viable product, went through multiple iterations of user testing and feedback, and arrived at a final game that we can be proud of. Through our demos, we have seen that the game is engaging to our target audience, so we can be sure we have fulfilled the requirements of our client. We believe that this game will become a successful exhibit in the museum.