# Science App for Lewisburg Children's Museum

Michael Hammer, Kenny Rader, and Keyi Zhang
Lewisburg Children's Museum, Prof. Erin Jablonski
*Department of Computer Science, Bucknell University, Lewisburg, Pa.*

## Abstract

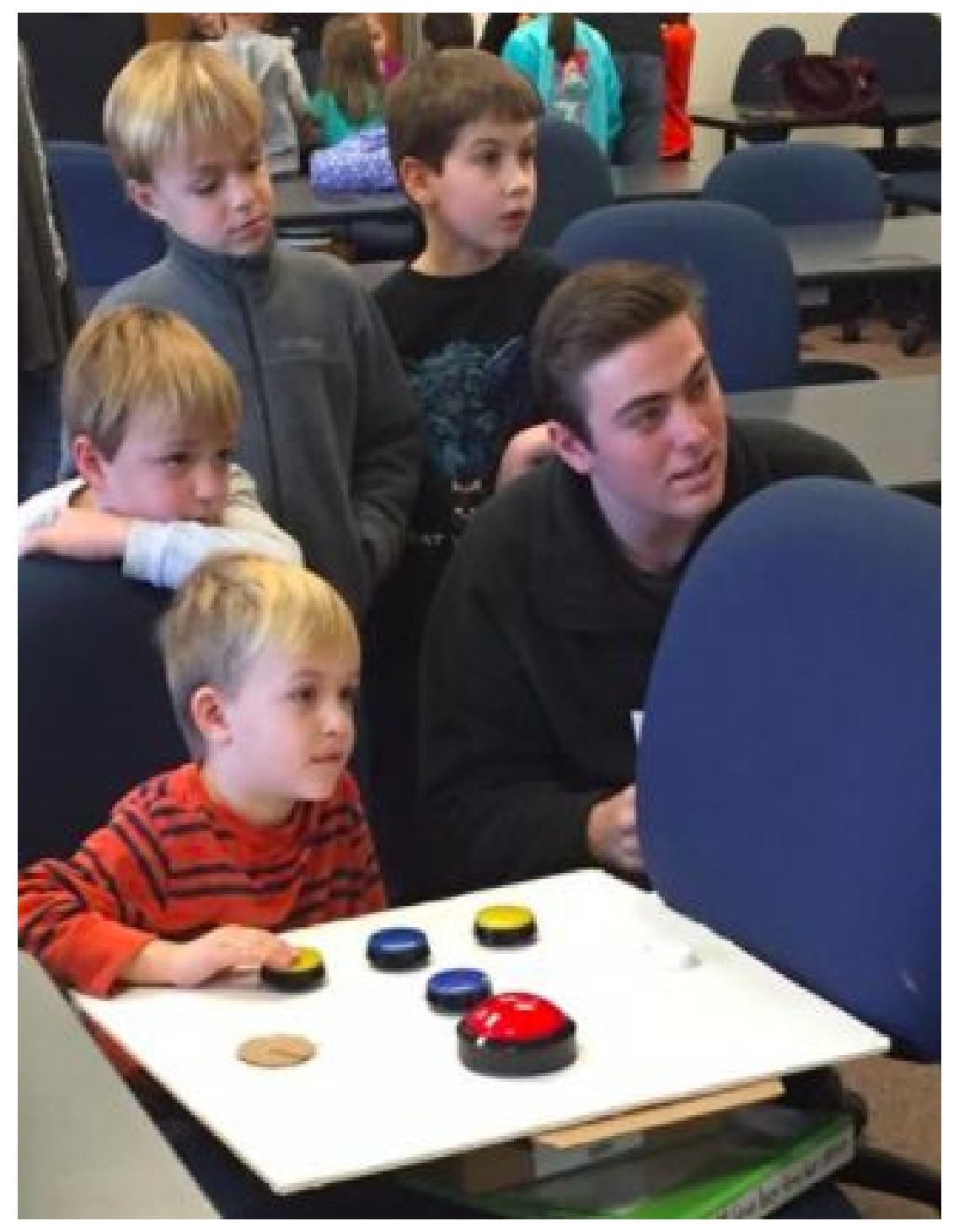
As technology evolves rapidly around us, it is important to keep children informed about science and technology so that they can be prepared for the life they are growing into. This can be challenging due to the complex nature of technology and what young children can comprehend. The Lewisburg Children's Museum is doing their part to resolve this dilemma by educating children about science and technology through their new space exhibit. Interactive learning is one of the best solutions to this task, combining the values of entertainment and education in one unique experience. Our team has been dedicated to designing a game that allows children to launch a rocket, teaching them concepts such as gravity, trajectory, and fuel use in a fun environment. To implement the game we used a variety of tools such as the Unity Game Engine and a custom-built analog controller. The game supports an easy to use interface and realistic graphics, as well as a physical control panel with interactive components and lights. Every aspect of the game has been engineered with the purpose of keeping the children engaged so that the can better learn these fundamentals of rocket launches.

## Background

**Goals:**

- Our main goal was to create an engaging, interactive environment in which children could learn about the fundamental concepts of projectile dynamics. Children learn much better when they can play with objects and devices and see how their input affects the world around them.
- Another goal was to make this project easily modifiable and accessible for future work. If, somewhere down the road, more functionality was desired of the system, this could be added with relative ease.

In order to leverage this "learning by curiosity", we needed to have an intuitive design for the analog controller that the kids would use. We narrowed down our design space through user tests, like the one on the right, to see how the children intuitively used the system, and how we could design around that.



## System Design

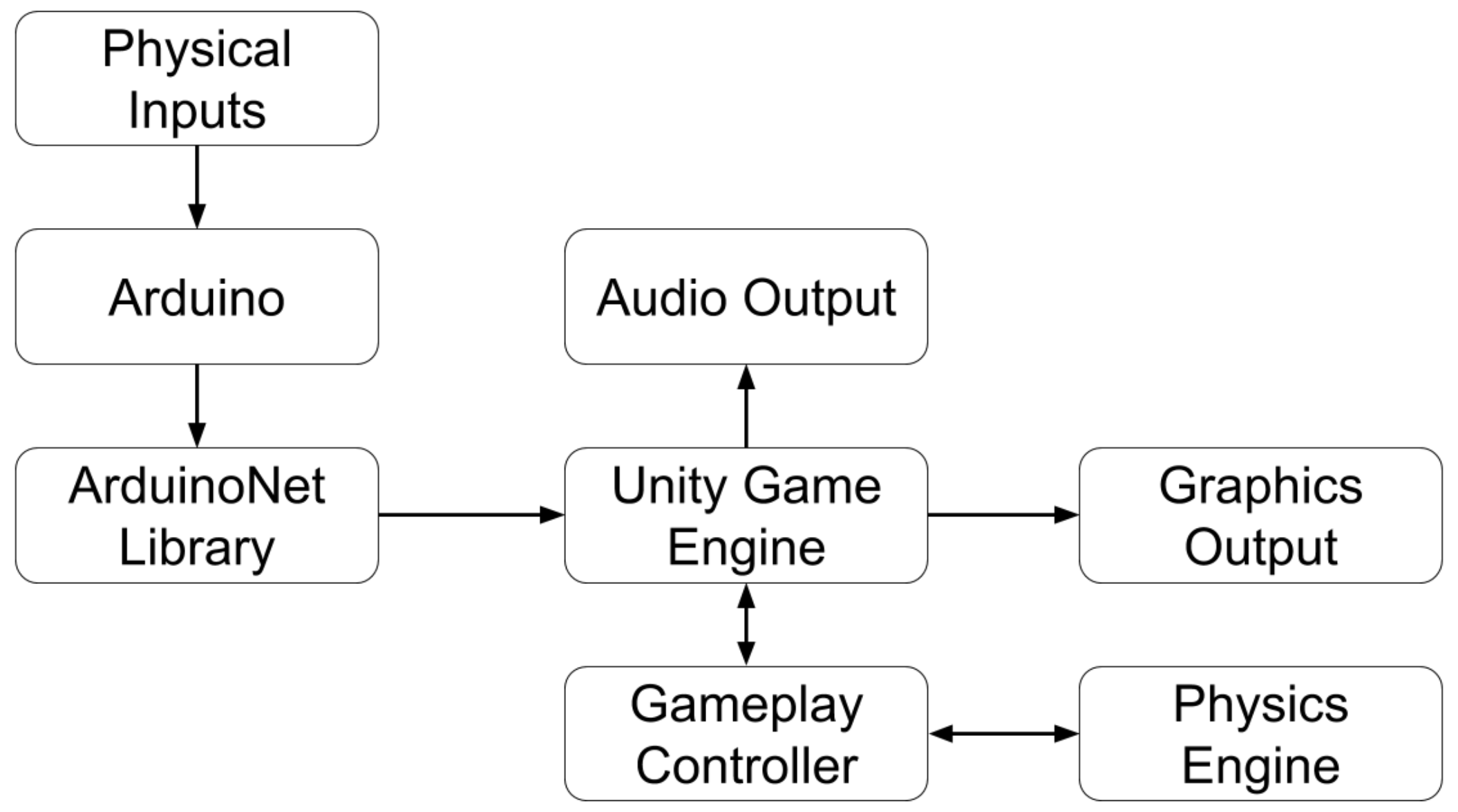


Figure 1: System Diagram

**Physical Inputs:**
This is a custom-built analog game controller that allows kids to easily interact with the game.

**Arduino & ArduinoNet Library:**
An Arduino was used to allow our analog controller to interface with our game software via a simple USB connection.

**Unity Game Engine:**
The Unity framework provided us with a platform upon which to develop scripts that control the game logic and display meaningful visual output.

**Gameplay Controller:**
Scripts handle controlling the game logic based on the inputs from the controller and the internal state of the game.

**Physics Engine:**
Physics scripts specifically handle the mechanics of the rocket's trajectory based on the user inputs, namely angle and fuel.

**Graphics Output:**
A monitor displays the visual output we developed in Unity. This is essential to the educational aspect of the game.

**Audio Output:**
Speakers relay sound effects for the game as well as voice-overs for children who are illiterate.

## Results

We conducted user tests to make the hardware layout as intuitive as possible.

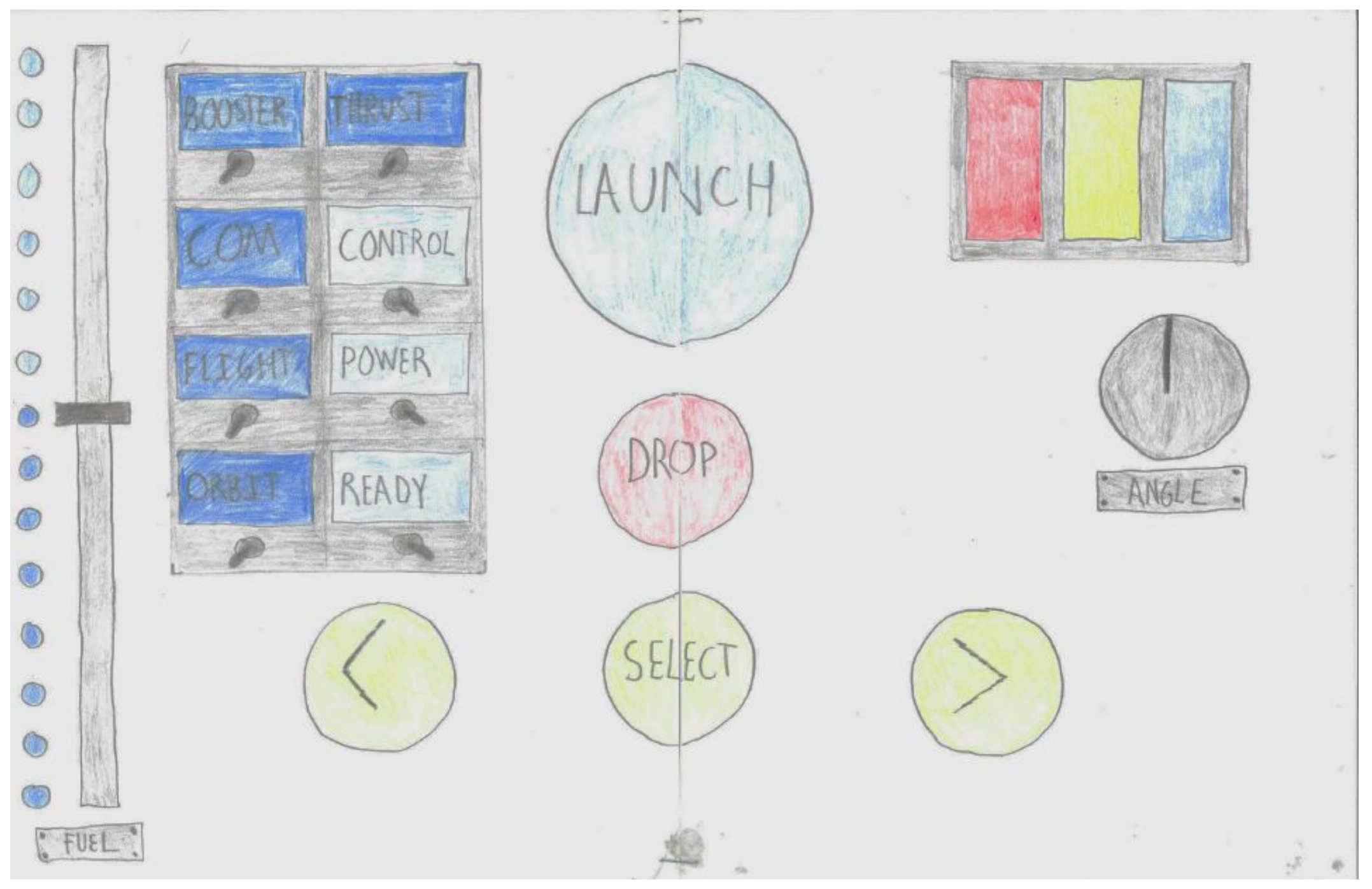


FIgure 2: Draft of first hardware layout design

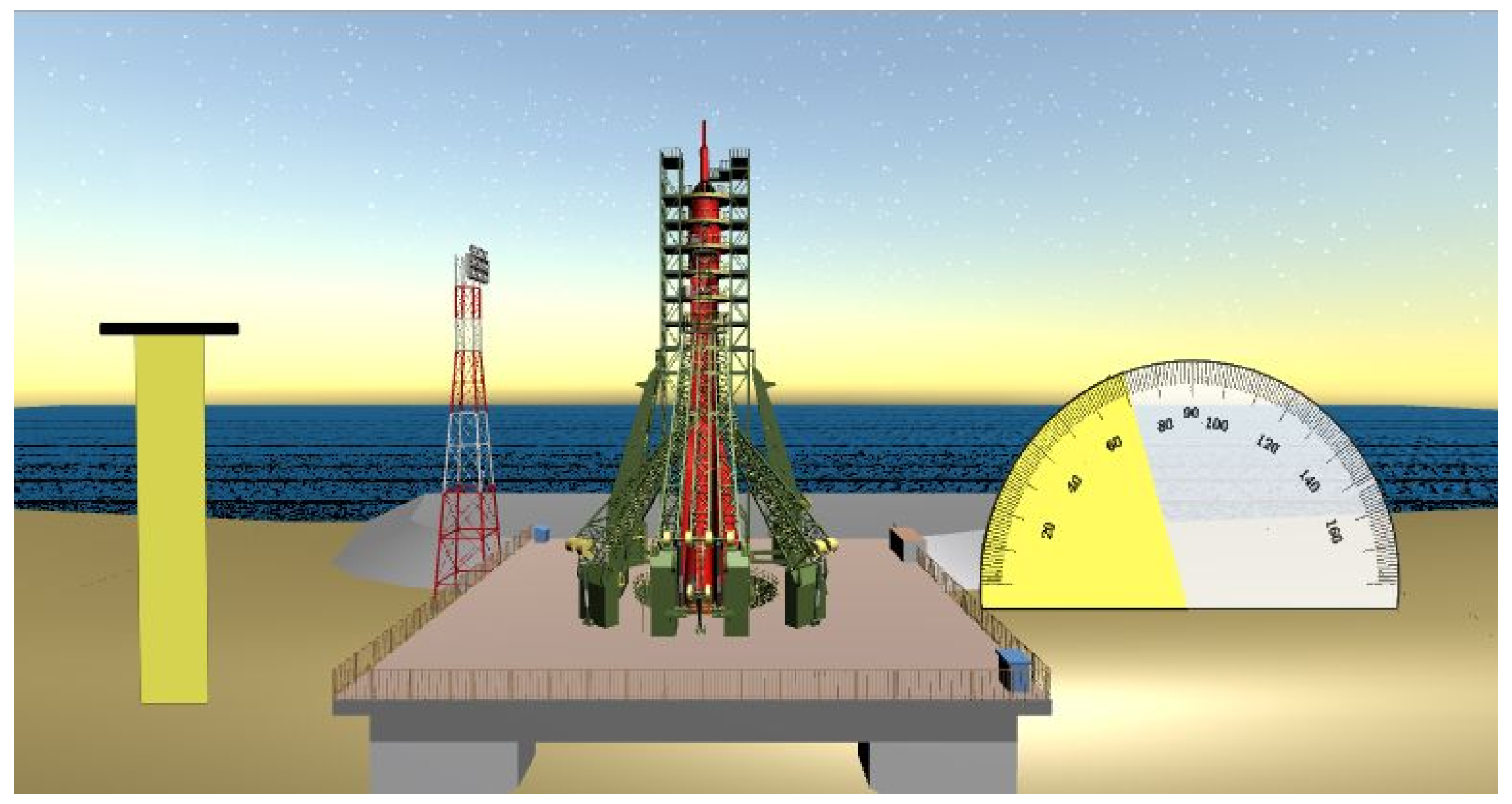
**User Test:**
The first iteration of our controller design features:

- A prominent launch button, and a few smaller buttons to control different aspects of the game's UI, including a fuel slider and an angle knob.
- Other decorative hardware components that are not functional in the context of the game, but are fun to look at.

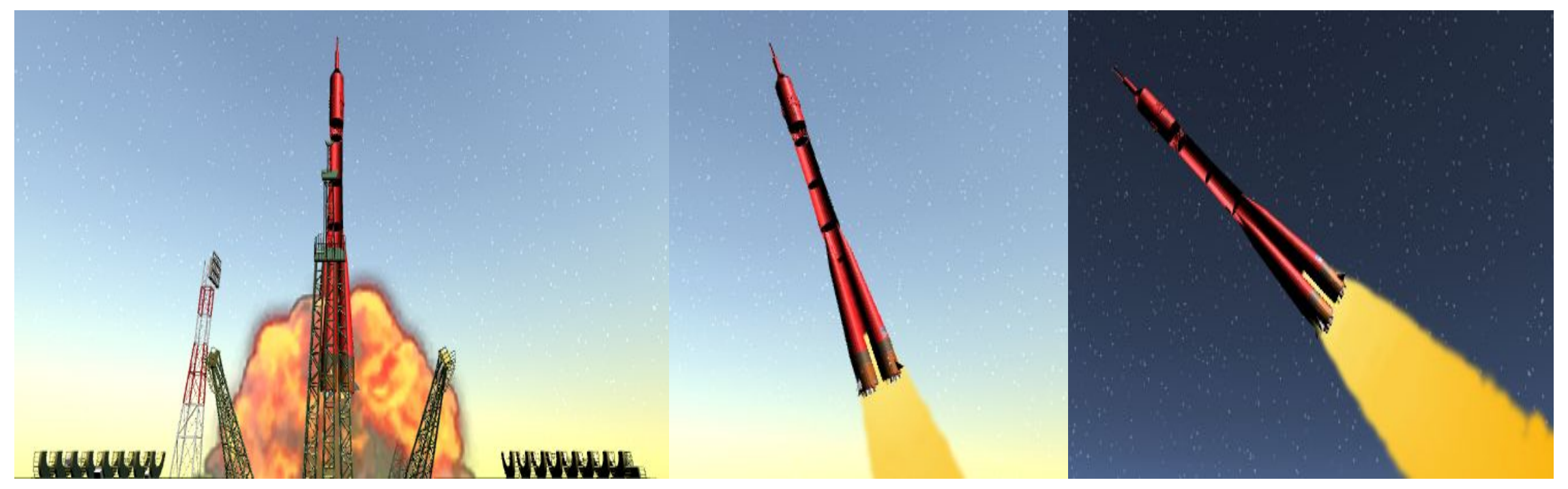
After making some early preliminary changes, such as removing the decorative components in favor of a sleeker look, we conducted the first user test.

The results of our redesign are featured to the right. The most notable changes that made the controller more intuitive and engaging are as follows:

- Paring down the number of important buttons, and placing them semi-radially around the launch button
- Adding color-changing buttons. These new buttons, located at the top of the controller away from the important buttons in the center, change the color of the rocket.



- The left slider: how much fuel the rocket will have
- The right protractor: the angle the rocket will turn to after launch.



- Launch phase: rocket travels directly up
- Turning phase: rocket rotates to the user's angle
- Trajectory phase: rocket follows a projectile path

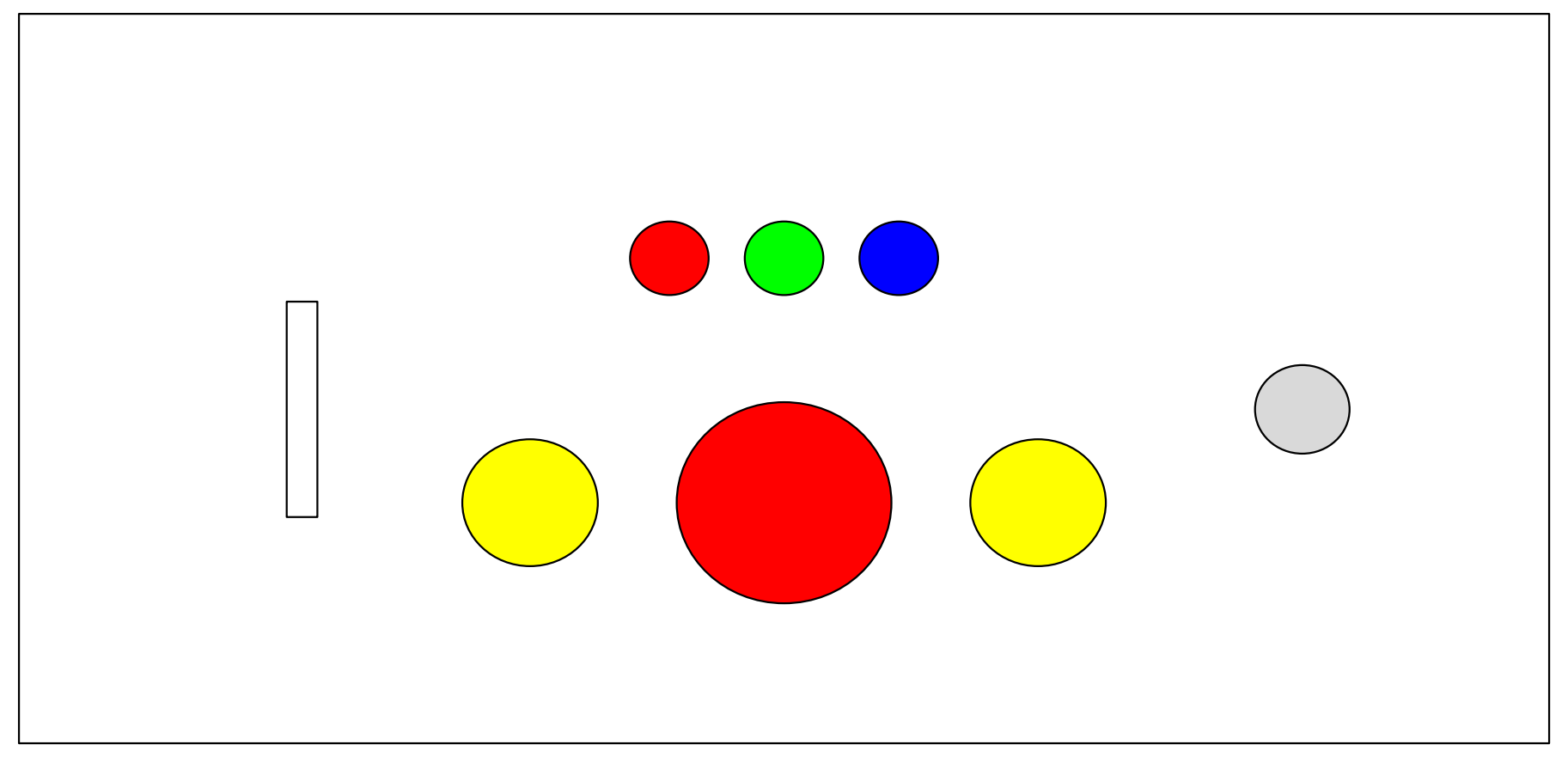


Figure 3: Final hardware layout design

## References

- lynda.com: Helpful video tutorials for Unity
- unity3d.com: Unity tutorials for more advanced users
- Professors Brian Utter and Adam Piggott for physics and mathematical modeling
- Professor Erin Jablonski and Paden Troxell for design direction and aesthetic advice