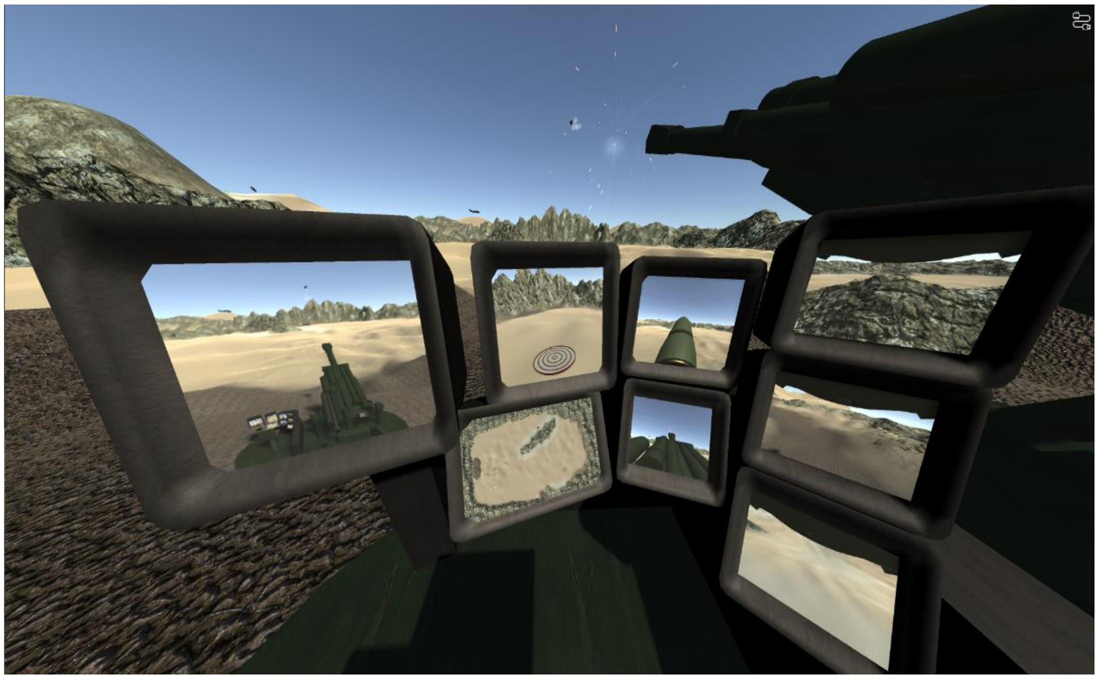
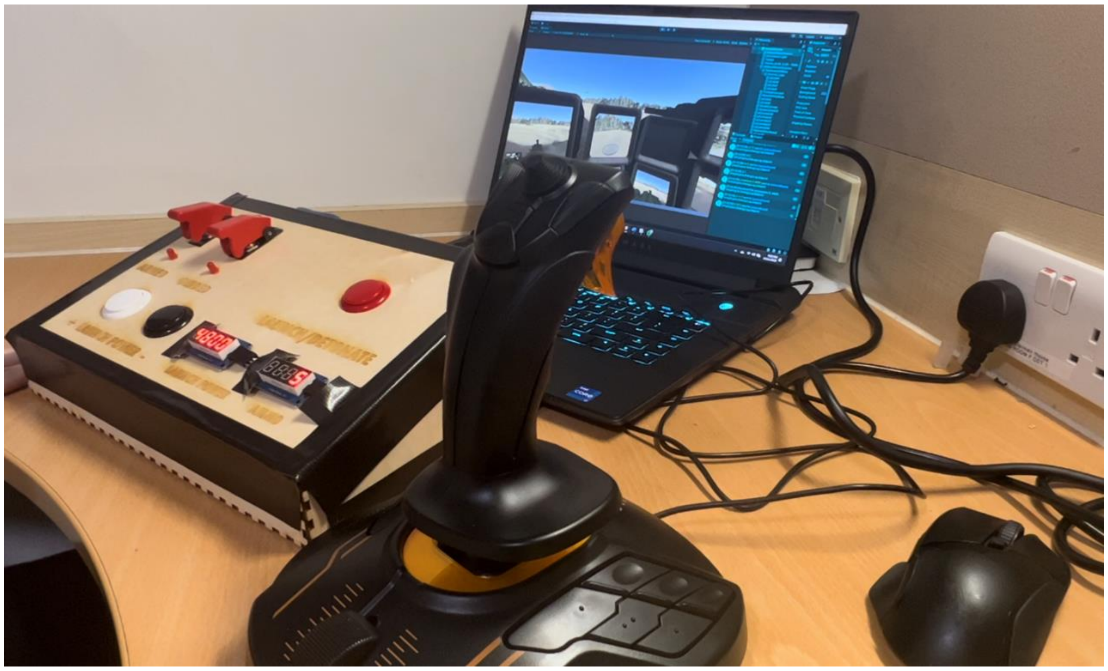




Artillery Rocket Simulator

By: Nathan Thomas-Benke

A pilot joystick along with a custom-built Arduino controller used to interact with a virtual Artillery/Rocket simulator built in Unity.

DEMO: https://vimeo.com/893496841

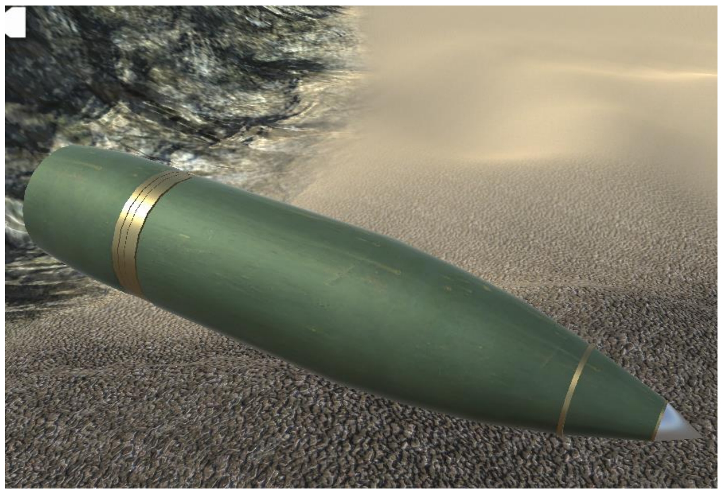
Introduction

With this project, the participant can use various interactable to alter functionality of projectiles being launched as well as switching between different modes within the simulation. The experience has been built with sound effects to make the participants feel immersed as they aim to hit static and dynamic targets located throughout the course.

Concept and Background Research

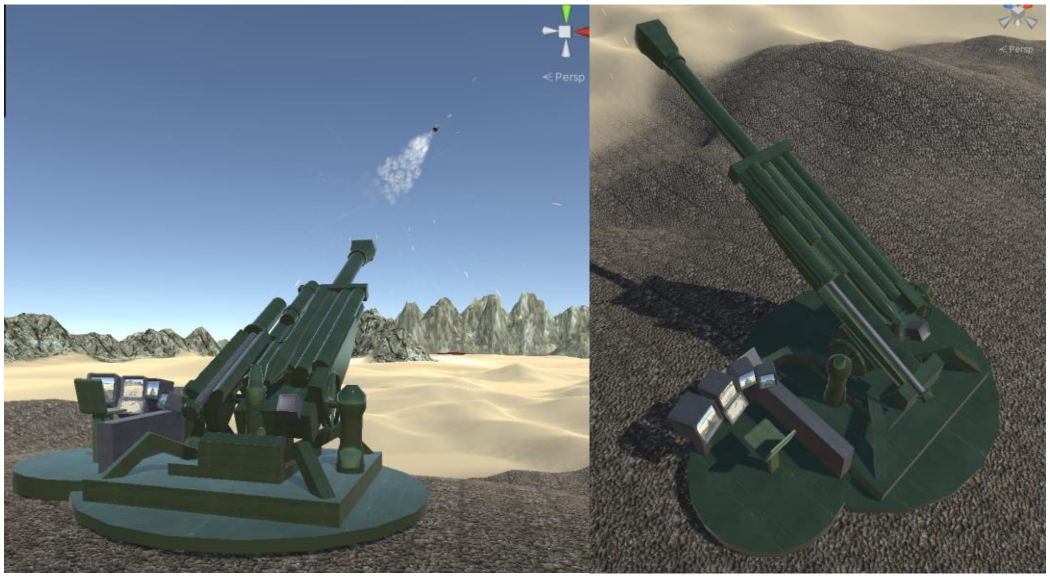
When coming up with a concept for the project, I wanted to create something that could have real-world applications while also having some relation to the VR/AR space. I knew I wanted the experience to be highly interactive which first brought me to wanting to create a flight simulator. However, due to my lack of knowledge in aeronautics, I instead shifted my concept into being an artillery simulator as it would require less degrees of freedom to be monitored as output. By watching footage of real Howitzer artillery being fired, I was able to get a general sense of the mechanics and more so, the scaling needed to launch the artillery shell. I then started looking at reference images of Howitzer artillery and its ammunition shells.

Reference Image (Howitzer Shell)



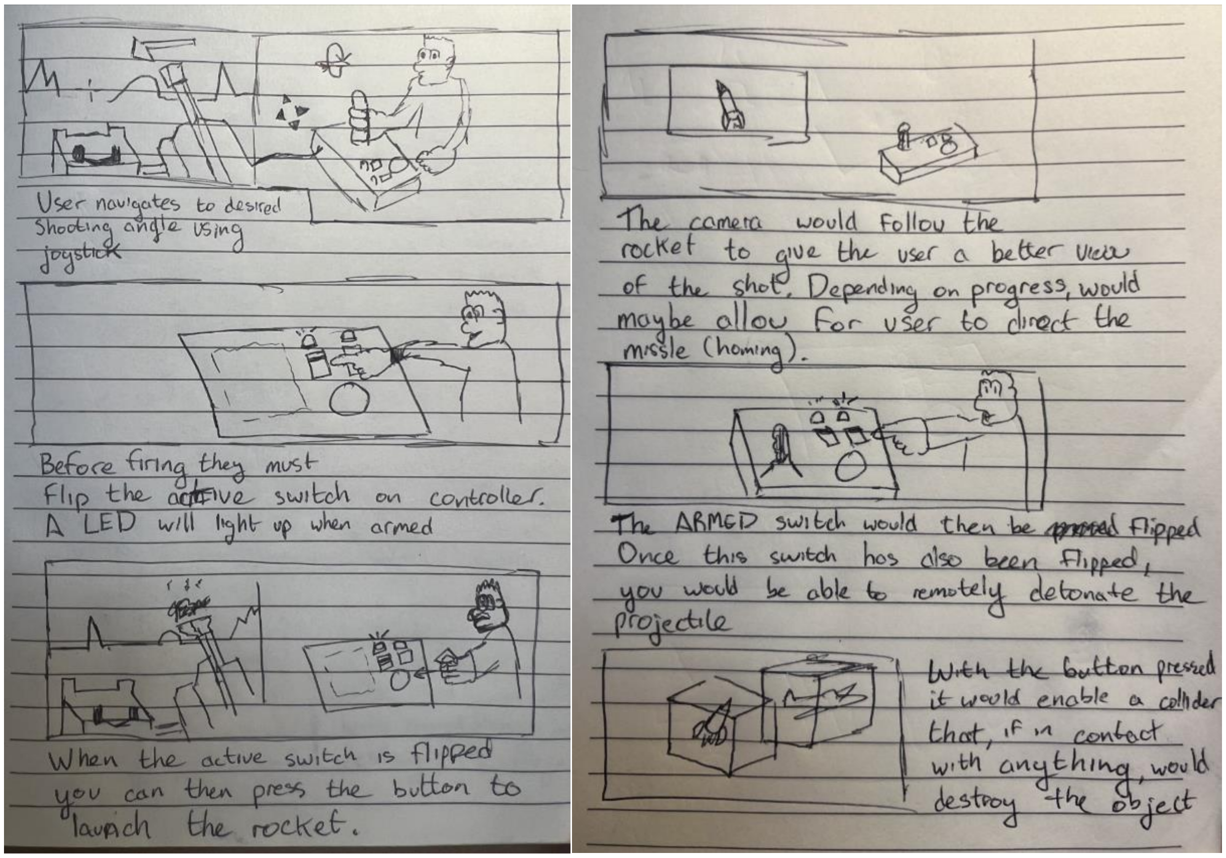
Modeled Recreation

Reference Image (Howizer Artillery)

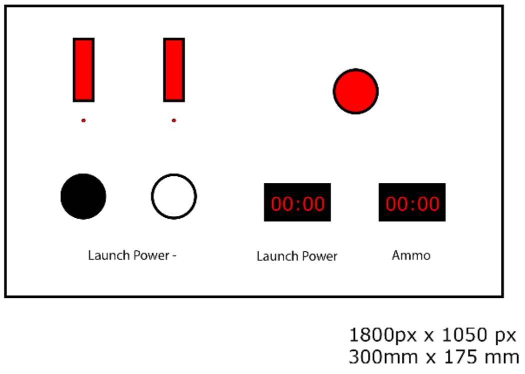


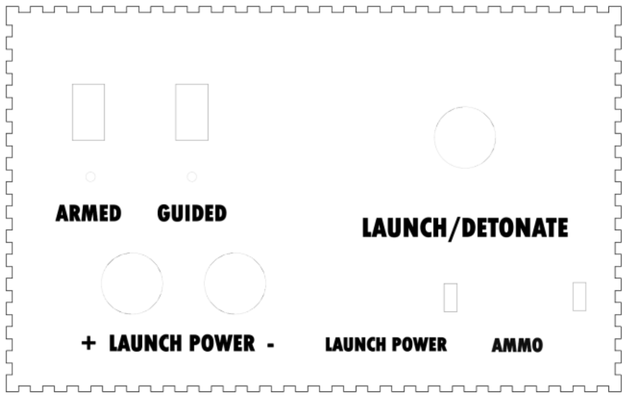
Modeled Recreation

An artillery cannon is essentially just a powerful cannon. This meant that all that the physics behind the launches was essentially calculating how powerful the shell was going to be launched and then applying that force in the direction that the artillery is facing. However, I felt that only having this simple interaction would cause the experience to not feel immersive as much of the time would be waiting for the rocket to land. This brought me to the decision to also implement a guided missile system, which would allow the user to take control of the rocket and its flight path. The physics system used to allow for this control was more challenging to establish due to the propulsion in which a rocket had.


User navigates to desired Shooting angle using joystick
Before firing they must Flip the ~~act~~ive switch on controller. A LED will light up when armed
When the active switch is flipped you can then press the button to launch the rocket.
The camera would follow the rocket to give the user a better view of the shot. Depending on progress, would maybe allow for user to direct the missile (homing).
The ARMED switch would then be ~~pressed~~ Flipped Once this switch has also been Flipped, you would be able to remotely detonate the projectile
With the button pressed it would enable a collider that, if in contact with anything, would destroy the object

Early Mockup of Functionality

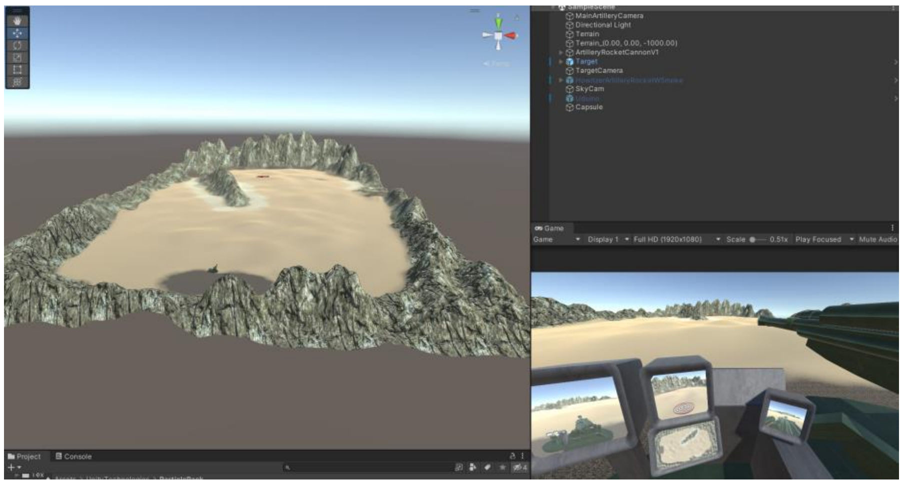




Controller Mockup *Illustrator Mockup*

Technical Implementation

I started off by creating the simulation portion as I researched and waited for the joystick and parts for the controller. This was also done to help find if and/or where there were going to be challenges with recreating different aspects of the simulation. Before attempting to implement the Arduino with Unity, I created a version in which you were able to maneuver and launch missiles with the keyboard.



Early version of the simulation environment

At this point, I had selected all the components for the controller and was able to have them delivered. Please see the parts list for better descriptions.

List of Parts Used:

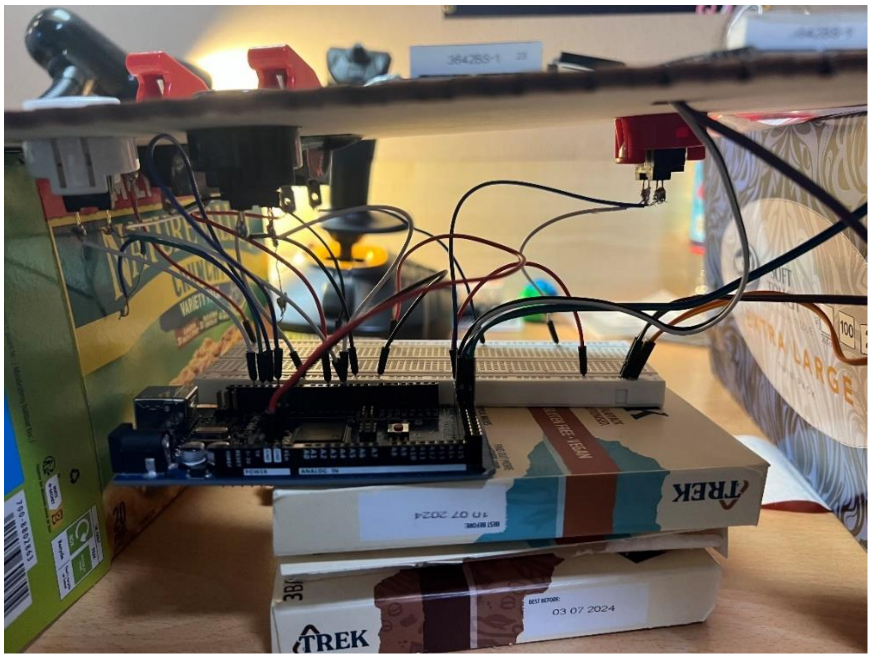
1 x Arduino Mega 2560	2 x 220 Ohm resistors
2 x Red LED	1 x ELEGOO Solderless Breadboard
17 x Breadboard Jumper Wire	8 x Female-to-Male Dupont Wire
1 x Thrustmaster T1600M FCS	2 x Hailege 0.36" 4-Digit Tube LED Segment Display Module. TM1637
3 x SANWA OBSF-30 Original Push Button 30mm	2 x Heschen Metal Toggle Switch Flick Flip

Build Tools:

1 x Gorilla Super Glue Gel	1 x 48mm Gorilla Tape Black
1 x 4mm Plywood panel (Hatchlab)	Epilog Laser Fusion Pro (Hatchlab)

The first external part I attempted to implement was the Thrustmaster joystick. This was much harder than I had anticipated due to it requiring a custom Input System within Unity that would overwrite the old input system which caused conflicts when attempting to augment it with my keyboard simulation. This was where I had to abandon implementing the hydraulic sound effects whenever the artillery cannon was moved, this was due to the differences in which the keyboard and joystick input data could be read by Unity.

Next, I began to implement the parts of external Arduino controller which included the 2 LEDs, the 3 push buttons, and 2 toggle switches. To do so, I used Uduino, a custom asset built to allow Unity to communicate with Arduino. To implement these parts into a controller, I needed to solder wires to each of the components so that they would be able to reach the Arduino or breadboard.

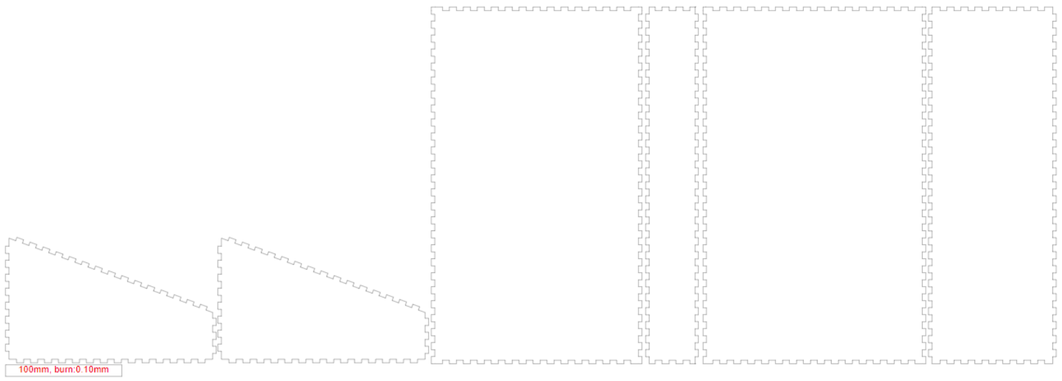


Makeshift workstation for wiring controller components before being placed in box.

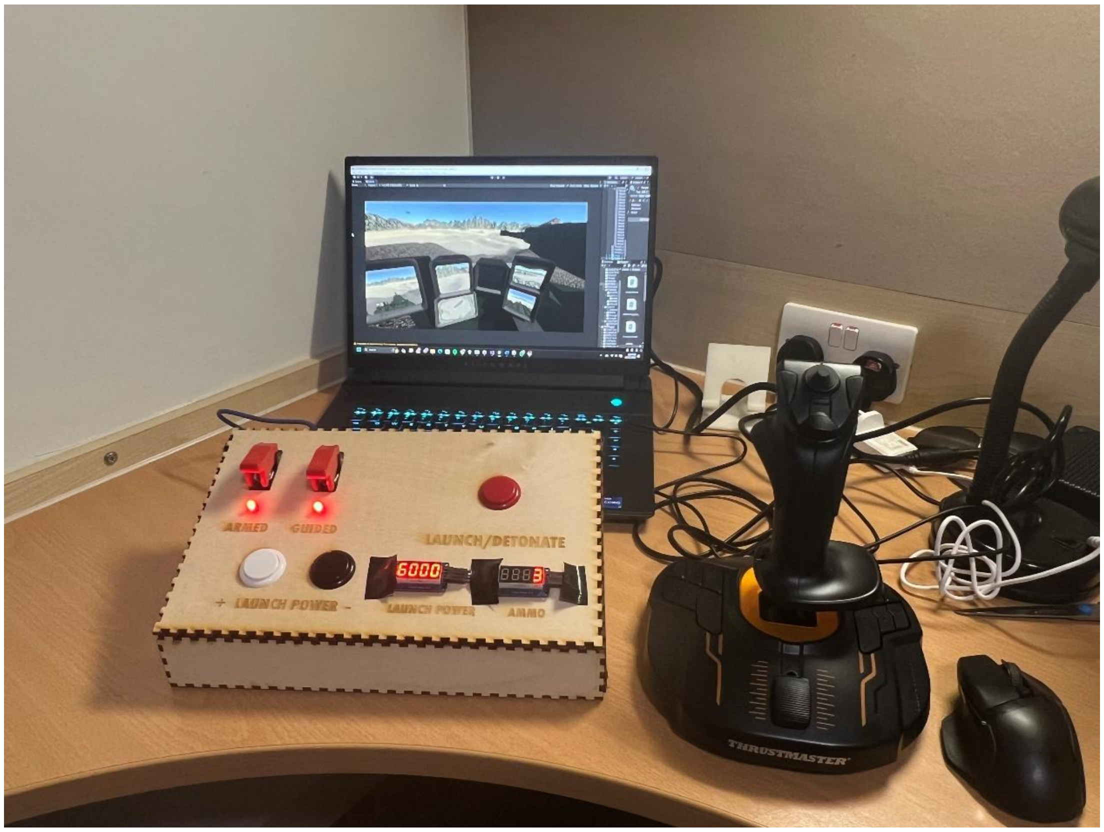
At this point, I implemented the guided missile feature within Unity which would allow for the user to take control and maneuver the launched missile using the joystick. This implementation was difficult due to the way in which physics needed to be applied to the shell. I made the missile more reactive due to the scale of the simulation. I also had to manipulate physics to allow for a sensible transition between an artillery shell and propelled missile.

Once these were implemented, I then began working to implement the TM1637 LED Displays. This was by far the most challenging element to get working as I had to create a way for Unity to send information back into the Arduino due to the LED Displays using a custom TM1637 library existing within the Arduino IDE. At one point, I nearly fried my motherboard due to the output from the TM1637 and Arduino as it was attempting to send data to Unity. From this, I went on to combine the TM1627Display and Uduino scripts so that they would be able to send and receive data from Unity without the motherboard stress issue.

After this, it was time to finish creating a container for the Arduino and input systems. I went through 2 versions in cardboard before moving onto creating a 4mm plywood version. The file was created by adapting a Console template from festi.info. Using the Epilog Laser Fusion Pro with Illustrator, I was able to create and cut a shell for the input components.

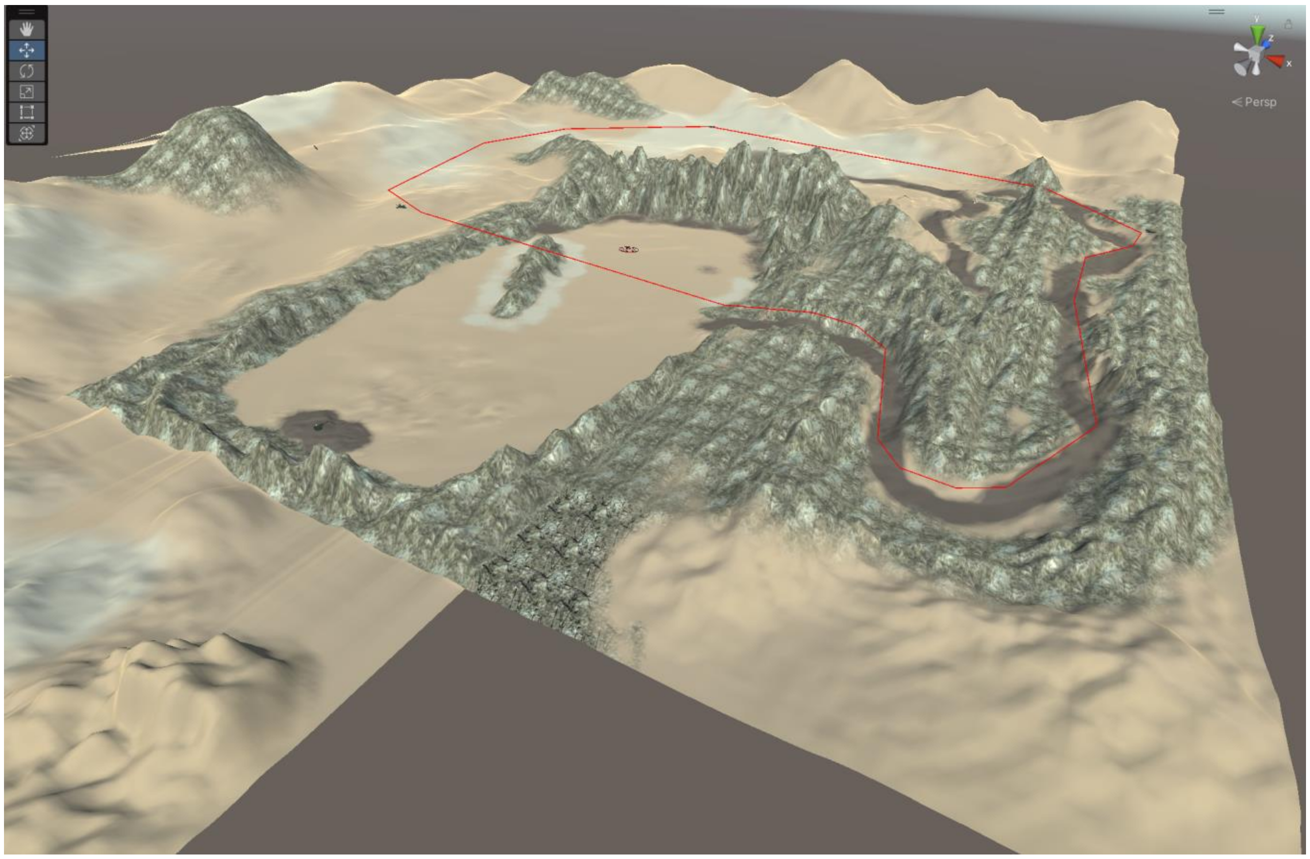


Adapted default file generated from festi.info

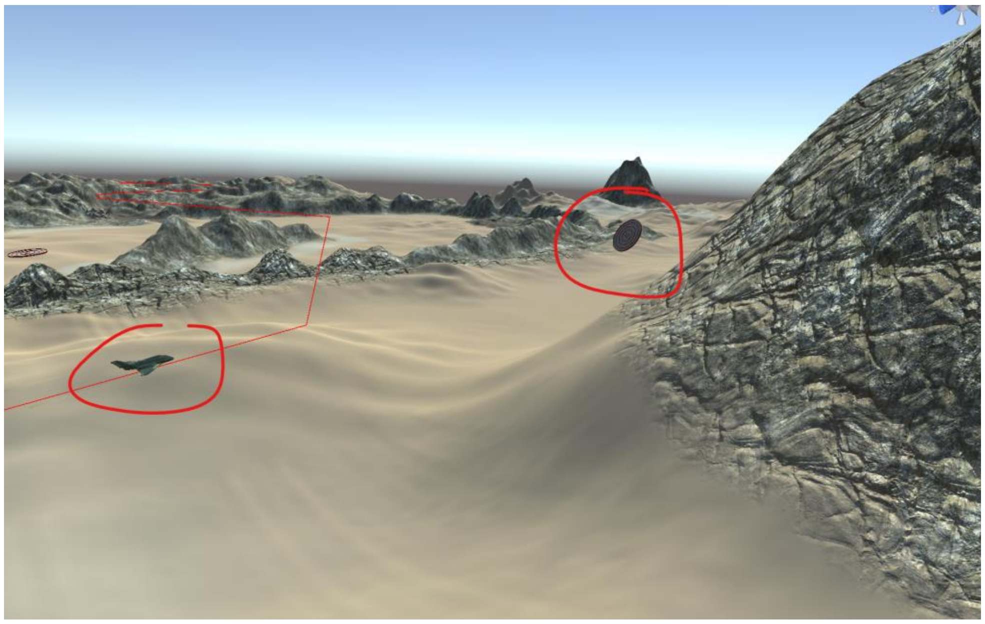


Final input systems setup with joystick and plywood Arduino controller

Once the controller was complete, I decided that it was important to add more goals for the user within the simulator. This is what led me to expanding the map to over triple its original size. I added additional targets throughout the map that the player could aim to hit and/or destroy. I also decided to use jet models (existing in Unity's Standard Asset) as dynamic moving targets. I implemented new scripts which allowed for these objects to fly along a circuit of waypoints. I also added code that would allow for the jet and small target objects to be destructible whenever a missile was remotely detonated within its vicinity.



Final extended map with flight circuit for jets



Additional destructible dynamic and static targets when added

Reflection and Future Development

Overall, I am happy with how the development of the project went. I felt that the way in which I iteratively implemented and updated the Unity simulation was effective as it gave me a way to control the simulation without the external controllers which was very useful when bug testing. My biggest struggle during development was the technical side of building the controller shell along with soldering the components. If I were to develop this project further, I'd want to expand the simulation to support larger environments and possibly even implement a version to where the player is situated as a gunner on an aircraft. This would call for further interactable on the controller. Alternatively, due to my pursuit of creating something that could be related to the AR/VR space, adapting the simulator into hybrid reality would be an exciting next step. This would call for creating a digital twin of the controller recreated within Unity and would omit the requirement for the mouse. With hand tracking, the simulator would be able to become highly immersive, which was a goal for this project.

References

Reference Images:

"155 Mm Caliber." *Wikipedia*, Wikimedia Foundation, 26 Nov. 2023, en.wikipedia.org/wiki/155_mm_caliber.

"Army Says It Needs $3 Billion for Artillery Rounds and Production." *NBCNews.Com*, NBCUniversal News Group, 8 Nov. 2023, www.nbcnews.com/news/us-news/us-army-says-needs-3-billion-artillery-rounds-production-rcna124137.

"Lightweight 155 Mm Howitzer System (LW155)." *USAASC*, asc.army.mil/web/portfolio-item/peo-ammo-lw155/. Accessed 11 Dec. 2023.

Reference Footage:

"US Marines Firing the M777 155mm Howitzer." *YouTube*, AiirSource Military, 24 May 2014, https://www.youtube.com/watch?v=IDM_zqkpK1U.

Tutorial:

Cannon Tutorial: https://www.youtube.com/watch?v=vTQuDgVBOYE

- Used in early concept of artillery cannon physics

Smoke Trail: https://www.youtube.com/watch?v=0mvaFKTTeUs

- Used in early concept of rocket smoke trails

Waypoint Circuit: https://www.youtube.com/watch?v=EwHiMQ3jdHw

- Used as basis of coding jet circuit pathing.

Documentation:

https://marcteyssier.com/uduino/docs/

- Adapted example templates for Unity and Arduino.

https://lastminuteengineers.com/tm1637-arduino-tutorial/

- Used for researching TM1637 functionality

Sound Effects:

Desert Ambiance: https://freesound.org/people/Imjeax/sounds/427401/

Hydraulics (Keyboard Version) : https://pixabay.com/sound-effects/homemade-hydraulic-hoist-30527/

Heavy Explosion: https://www.zapsplat.com/music/heavy-impact-sound-of-a-artillery-firing-close/

Distant Explosion: https://www.zapsplat.com/music/explosion-distant-1/

Artillery Reload: https://www.pond5.com/sound-effects/item/66799142-foley-various-foley-atrillery-shell-load-magazine-cannon-dee

Unity Assets:

Uduino: https://assetstore.unity.com/packages/tools/input-management/uduino-arduino-communication-simple-fast-and-stable-78402

- Used for connecting Uduino and Unity

Unity Standard Assets: https://assetstore.unity.com/packages/essentials/starterassets-firstperson-updates-in-new-charactercontroller-pac-196525

- Used Jet Prefab

Unity Particle Pack: https://assetstore.unity.com/packages/vfx/particles/particle-pack-127325

- Used Explosion Prefab