

# Project Deliverable F - Group 5

## Prototype I and Customer Feedback

GNG1103-C01

November 4th, 2023

Noor Trigui 300350022

Ava Butts 300350306

Luca Chayer 300375838

Dev de Haan-Sharma 300339994

### Table of Contents

Introduction.....	2
Client Meeting Recap.....	2
Wrike Link.....	3
Prototype 1 and test.....	3
Analysis of critical components or systems.....	3
Explanation of code.....	3
Diagram of animations.....	3
Explanation of diagram.....	4
Video of door loop (DOUBLE CLICK TO VIEW).....	4
Prototyping test plan.....	5
Test plan from Deliverable E:.....	5
Qualitative observations.....	5
User feedback.....	5
Feedback of user 1:.....	6
Feedback of user 2:.....	6
Feedback of user 3:.....	6
Current State of Tests.....	7
Overview of Prototypes and Tests.....	8
Conclusion.....	10

## Introduction

Throughout the one-on-one client meeting with Mines Action Canada, we were able to share our ideas that we had generated in the Ideate stage. Afterwards, we decided to alter our project so that it aligned more with the client feedback and do some more research to ensure that our VR simulation effectively showed how life would change in the presence of killer robots. One of the key takeaways from this research was the importance of the message of dehumanization (possibilities for hacking and racial profiling) in our VR; this was one of the main themes that were brought up in each source that we looked at. When planning our updated project in Deliverable E, we included elements that would show effects of dehumanization in a user's home such as a radio with patrol times or a sign that would remind the family to put their masks on. The project development has three goals: interactivity, movement, and realism. As a group, we decided that interactivity was the most crucial goal to tackle at this point in the design process so we worked on Test ID 4 by coding the open-close loop of the door, which would align with scene 3 in our storyboard. This was a focussed prototype for one of the key scenes in our project. It is also one of the more difficult interactions in our project so any future interactions we need should be easier to code now that we know the general functions. After completing the code, we also got feedback from three users which can help us decide what to focus on for our higher fidelity, final product.

## Client Meeting Recap

After the second client meeting with Mines Actions Canada we were able to gain insight on what the client liked and disliked about our design idea as well as possible improvements we can make to our final design. After discussion it was determined that the client did not want to have the simulation be in the point of view of the robot as it gives off the wrong impression. The client stated that when you enter a simulation you embody the role you are playing and due to that becoming the robot portrays a negative message. Jason had stated that if it is in the point of view of the robot the user will become sympathetic to the robot instead of the people being affected by them which in turn is displaying the wrong message. The client loved the idea of having the simulation run through a suburban environment as it was interesting and relatable. However, they stated that it must be simple while still getting the specific message across on how humans have adapted and why these weapons should be prohibited. The client mentioned how having the warning signs and all working adaptations such as the timed patrols will make the simulation more convincing and will set the mood for the overall message. A main point that was brought up several times was how they want the simulation to relate to ordinary people specifically in Canada but should still be able to translate to the rest of the world. This is because our main objective is to convince policymakers, specifically Justin Trudeau to prohibit autonomous killer robots. Finally, the client had mentioned it was extremely important to do further research into autonomous killer robots through websites such as the International committee of the red cross, remote controlled war code and the immoral code documentary in order to help us further understand the clients needs and focus on the kind of failures these types of robots have and how it will be detrimental to society.

## Wrike Link

<https://www.wrike.com/frontend/ganttchart/index.html?snapshotId=f6f0i14vu66feGUY4GuOdaehcD0ceGmt%7CIE2DSNZVHA2DELSTGIYA>

## Prototype 1 and test

We decided that the first prototype would be to “Ensure objects can be interacted with (Test ID 4, Test Plan Deliverable E)” as it is the most critical part of our final design. For this prototype we decided to design the door for the house. The door will be openable with a mouse click and will close again with another mouse click.

## Analysis of critical components or systems

Here is the code for the door opening.

## Explanation of code

To start, the anim syntax grabs the animations we’ve made in unity. The get mouse down gathers input from the mouse. When the mouse is clicked the loop starts. The ray ray line gathers the position of the mouse and the raycast finds out if the click was on the object. From there the if-else loop determines the doorstate (1 being open 0 being closed). Once the doorstate is determined the correct animation is triggered and then the doorstate is altered.

## Diagram of animations

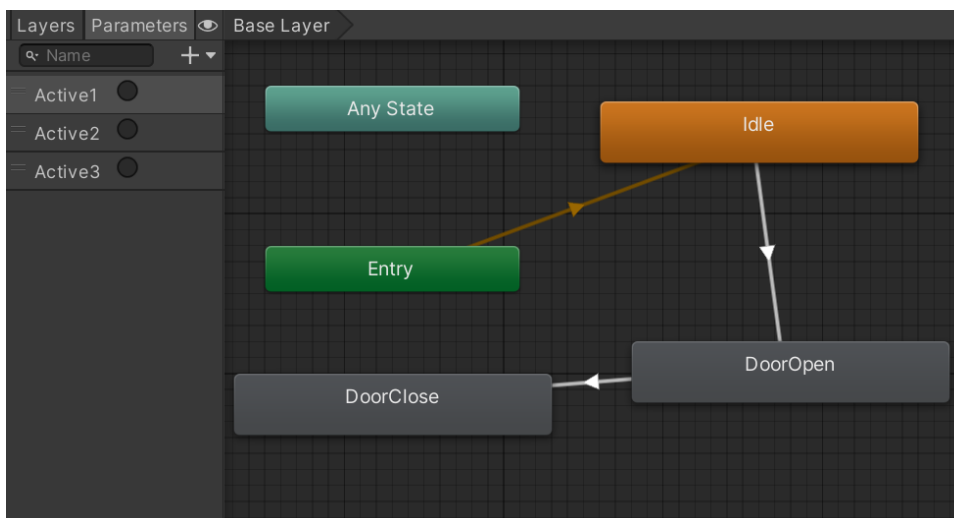
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class DoorOpen : MonoBehaviour
{
    Animator anim;
    private int doorstate;

    void Start()
    {
        anim = gameObject.GetComponent<Animator>();
        doorstate = 0;
    }

    void Update()
    {
        if (Input.GetMouseButtonDown(0))
        {
            if (doorstate == 1)
            {
                anim.SetTrigger("Active2");
                doorstate = 0;
            }

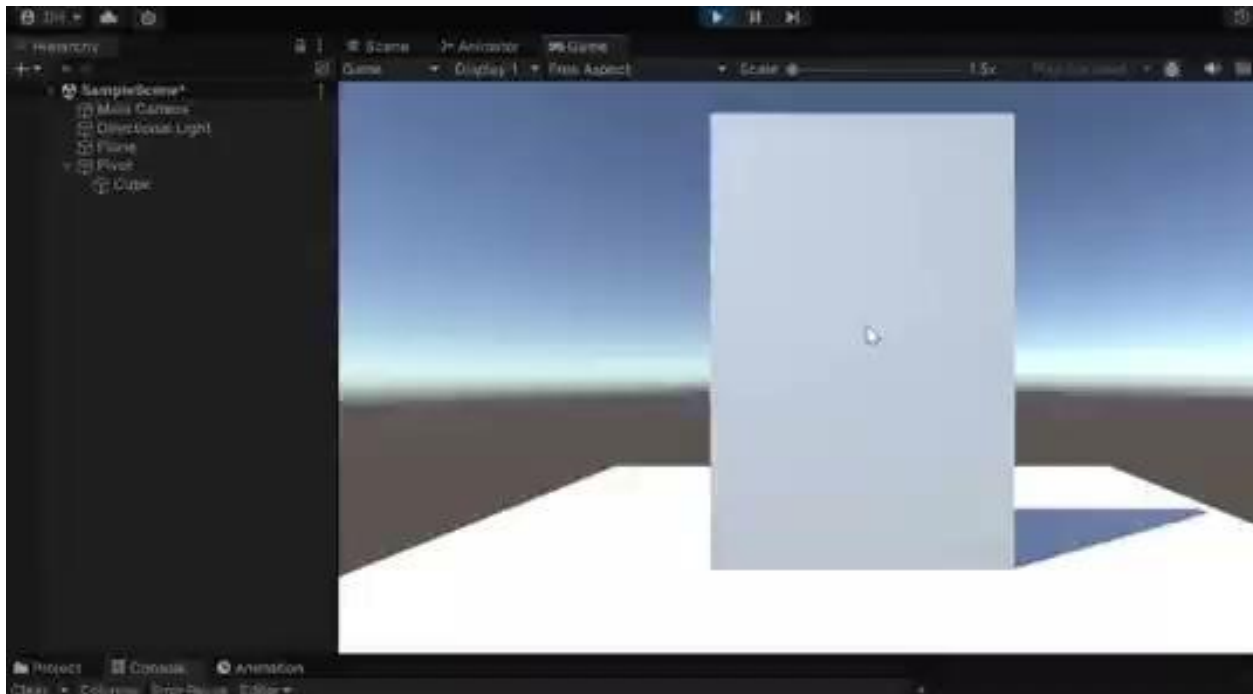
            Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
            RaycastHit hit;
            if (Physics.Raycast(ray, out hit, 100))
            {
                if (doorstate == 0)
                {
                    anim.SetTrigger("Active1");
                    doorstate = 1;
                }
                else if (doorstate == 1)
                {
                    anim.SetTrigger("Active2");
                    doorstate = 0;
                }
            }
        }
    }
}
```



## Explanation of diagram

Each of the boxes represents an animation which we have made in Unity. The vectors in between the boxes represent transitions so the animations can flow one into another. The transitions are dependent on triggers from the user and they are shown in the top left “Active1...” The triggers act as a loop. At first the door is closed but once the Active1 trigger is activated the animation starts and the door opens. Now that the door is open the system is waiting on another input from the user to trigger Active2. Once Active2 is triggered the door closes.

**Video of door loop (DOUBLE CLICK TO VIEW)**



**Prototyping test plan**

Once we completed our working prototype we had to test it based off of our test plan outlined in deliverable E.

**Test plan from Deliverable E:**

Test ID	Objective	Prototype & Test Description	Results to be Collected	Estimated Time
4	Ensure objects can be interacted with	Design a simple code that produces a response when the user interacts with an object. Run this code through an undeveloped	Qualitative observations on if the object responds to interaction as intended. Designate one clear change in	Given the simplicity of the test, this should only take an hour to develop and test.

		environment with a designated object to be interacted with.	behavior, such as change in color.	
--	--	-------------------------------------------------------------	------------------------------------	--

Because of the inherent consistency that code provides and the simplicity of our prototype we did not need to run many tests. We tested the door interactivity cycle 3 times and had the same results each time.

## Qualitative observations

Aspects of prototype that work as intended

- Door opens with a mouse click
- Door closes with a mouse click
- Door stays open as long as the user does not click
- Animation is smooth and lifelike

Prototype shortcomings

- The opening and closing door cycle cannot loop
- Door can only close when the width of the door is clicked

Change in behavior for next iteration

- Door can open and close as many times as the user wants
- Door can be closed when any portion of the door is clicked

## User feedback

While it would've been fun to go up to parliament and ask the parliamentarians to try out our door code as they are our ideal users of our final system we realized that this might not be the best use of our time. We instead decided to ask our friends and roommates to try the code as they probably have about as much experience in door unity codes as the people on parliament hill. We selected three users and gave them the laptop with the simulation running with no input or guidance from us. We monitored their actions and what they were saying out loud.

### Feedback of user 1:

Qualitative observations of User 1's experience

- User 1 immediately clicked on the door and it opened immediately
- User 1 then tried to close the door by clicking the same place as they did to open it and was unsuccessful.
- User 1 clicked on various parts of the door for roughly 15 seconds before clicking on the width of the door to close it again.

**User 1 Quote:** “Very intuitive to open the door. Closing it, not so much”

---

### **Feedback of user 2:**

Qualitative observations of User 2’s experience

- User 2 opened the door immediately like user 1
- User 2 was also confused as to how to close the door but figured it out by spam clicking until they clicked the right spot
- Once the door was closed user 2 tried to open it again but was unable to
- User 2 tried to open the door for roughly 10 seconds before giving up

**User 2 Quote:** “I wish I could open and close it again”

---

### **Feedback of user 3:**

Qualitative observations of User 3’s experience

- When User 3 was given the laptop they didn’t know what to click on or do
- User 3 sat dormant looking around the unity interface unsure of what to do.
- After about 45 seconds of inactivity user 3 clicked on the door and exclaimed “I didn’t know that was a door”
- Once user 3 opened the door they like User 1 and 2 couldn’t figure out how to close it for roughly 10 seconds until they figured it out

**User 3 Quote:** “Overwhelming”

## **Current State of Tests**

At this stage we have completed one prototype and test (Test ID 4). We will continue to follow our prototype test plan from deliverable E shown below in the future tests. The next tests that we will do would focus on movement and importing all the assets (Test ID 2, 3 & 5).

### **Overview of Prototypes and Tests**

<b>Test ID</b>	<b>Objective</b>	<b>Prototype &amp; Test Description</b>	<b>Results to be Collected</b>	<b>Estimated Time</b>
1	Ensure that the storyboard communicates	Make a basic storyboard which has screenshots of the	Qualitative observations based on other students, and hopefully the client,	About one hour to make the storyboard and a

	our main message	assets we are planning to use and communicates the intended game play and scene structure	of how well the story board communicates risks of autonomous killer robots	week to gather sufficient feedback / perspectives
2	Ensure the player can move properly	Set a list of paths meant to correspond to certain inputs, and run these using a “movement” script in an undeveloped environment.	Mostly qualitative observations of how well the program produces the desired movements.	Given we already have an idea of such a program, I assume it would take one hour to develop the script and test it.
3	Ensure that the scenery is realistic based on the assets used	Import the assets into the scene and make sure that the movement still works. Update the scene to make it more realistic and fit our original storyboard	Qualitative observations to check that the scene is realistic and assets are successfully imported	It will take 20 mins to import the needed assets but we may take an hour or more to play around with it and adjust the scene based on our research and user feedback
4	Ensure objects can be interacted with	Design a simple code that produces a response when the user interacts with an object. Run this code through an undeveloped environment with a designated object to be interacted with.	Qualitative observations on if the object responds to interaction as intended. Designate one clear change in behavior, such as change in color.	Given the simplicity of the test, this should only take an hour to develop and test.
5	Ensure that movement works in closed environments (the house where most of the VR will take place)	If the program from test 1 runs successfully, it can be tested in a closed environment, most likely whatever model for the house we end up using. The test would be similar using paths and inputs, not also causing collisions between the player and walls to see how	Same observations as the first movement test (Test ID = 2), with keen attention on how the player interacts with other objects.	Given the code would be written, this should take only 30 minutes.

		they interact.		
6	Adjust the house asset to show the effects of autonomous killer robots (see project plan for these elements)	Examine the window and radio (scene 1) and the window, masks, and signs (scene 2) making sure that they are realistic and visible no matter how you move around the scene	Qualitative observations to check that the scenes are realistic	Could easily take a couple hours to make adjustments to the surroundings and possibly even to the code of the VR
7	Ensure that the transition from the indoor to the outdoor environment is seamless and doesn't cause any lagging or discomfort	There must be a virtual door that users can interact with to exit the house as well as an animation of the door opening to reveal the outside environment (similar elements in every house - curtains over windows, locked doors, etc.). Some lighting changes would make the outdoor setting more realistic	Qualitative observations to check that the scenes are realistic and that the transition does not introduce any performance issues, such as lagging	May take about one hour to code and test the transition but some adjustments may need to be made to the outside world which would take a longer period of time (about an hour or two more)
8	Integrate sound effects into the VR based on certain interactions (ex. creaking noise when the door to outside opens and radio which would play throughout the VR)	Ensure that sound effects are in synch with VR interactions and that the sounds don't produce any lagging or affect user experience	Qualitative observations to check that the scenes are realistic and that the sound effects do not introduce any performance issues, such as lagging	Given the majority of the code would be written and most of the VR should be designed, this should take only 30 minutes to an hour.

### Conclusion

This report outlines the progress made in developing a VR simulation for Mines Action Canada. Based on the client feedback, we decided to put more of an emphasis in conveying the message of dehumanization and making sure that the user would feel sympathy for a person going through the experience as opposed to a robot patrolling a neighborhood. Through developing the door-opening mechanism, we learned some of the functions that we may need to code future interactions for this project. Testing results show that sometimes it was difficult to figure out how to open and close the door and there were challenges in



repeatability. Next steps for this project include importing our chosen assets, embedding the door interaction into the asset, and focussing on our second project development goal: movement. This would necessitate that the user moves smoothly within the bounds of our chosen assets.