Project Deliverable G - Group 5

Prototype II and Customer Feedback

GNG1103-C01
November 12th, 2023

Noor Trigui 300350022
Ava Butts 300350306
Luca Chayer 300375838
Dev de Haan-Sharma 300339994

Table of Contents

Introduction

In the ongoing development of our virtual reality simulation for Mines Action Canada, exploring the societal impacts of autonomous killer robots, our recent client meeting and pitch presentation reinforced our initial ideas and allowed us to reflect on our work so far. One of the main points brought up by the client was time concerns, and we responded by allocating more time to tackle coding challenges and work on prototypes that would build up to a minimal viable project; this would ensure that we address the client's problem and proposal by Design Day. Another adjustment was in our Bill of Materials where we removed a $20 mask asset that we thought would not benefit our design. Throughout the rest of the prototypes and tests that we do within the coming weeks, we intend to focus beyond technicalities especially as the client explicitly wants us to delve into issues and adjustments that may arise in the presence of autonomous killer robots. In previous deliverables, we have split our project development into three goals: Interactivity, Movement, and Realism. Our first prototype (Deliverable F) was centered around the goal of Interactivity as it focussed on the specific mechanism for opening and closing the door, which aligns with scene 3 of our storyboard. The second prototype focuses mainly on the goal of Movement through coding camera rotations and movement based on mouse/cursor movements and WASD keys. Given approval of our Bill of Materials, this movement was tested within the house asset intended for the final product which makes our testing more realistic.

Client Meeting Recap

During the client meeting not only did we gain feedback for our own prototype but we were also able to hear what feedback others gave to see if we could implement it into our own prototype. The question that was asked to every group that presented was if the coding process had started and whether or not after completing some of the code the final prototype would be feasible by design day. Although the coding took a long time to complete we believe it is still possible to have the final prototype done by design day. Before the client meeting we had created the code for the animation to open the door. At the moment, we currently have the asset that the simulation will take place in and have added a movement code to allow for the user to move around freely. If we feel like the simulation will not be ready by design day, or if we can't keep up with the project test plan, we know to focus on ensuring we're at least capturing the idea of how civilians have adapted to these autonomous killer robots and will only add the extra details if we know we have enough time. After the last client meeting we were told to do some research on autonomous killer robots/weapons and try to implement our findings into our final prototype. Through our research we were able to determine the underlying issues of these weapons, profiling and systemic injustices. The client liked the connection we had made between the underlying issue and the prototype. However, she said it's important to ensure we make it clear what part of the robot we're trying to tell the story about. She had also said that with the specific assets we have we are on track to tell our story. However, we need to ensure the story translates to everyone, especially people who don't have any background information on autonomous killer robots. This message is being translated through the VR simulation by having face masks and posters on the door reminding people that masks are mandatory in order to stay safe while exiting the house. During the client meeting, we had certain questions we had wanted answers to, one of them being if the concepts of racial profiling was going to be too sensitive of a topic for certain users and if we should caution against the idea of implementing it into our prototype. The client had said that there is no need to caution against it and to go forward with our prototype.

Updated BOM

Updated Bill Of Materias

Prototype Plan

Compared to our previous prototype, this one was meant to be more comprehensive, have a slightly higher fidelity, and use some of the assets that we intend to include in our final project. The interaction with the door in the prototype for Deliverable F was used as a proof of concept on a focussed scene of our storyboard to start to understand the coding and layout of Unity as well as to get feedback from some users and our clients. This next prototype is more comprehensive, addressing Test ID 2, 3, and 5 in our original prototype test plan. The main stopping criteria was to ensure that the user can walk around in a more refined environment with minimal bugs/lagging. Future prototypes will introduce some of the more interactive elements into our design including the radio, the door opening, and masks in the mudroom.

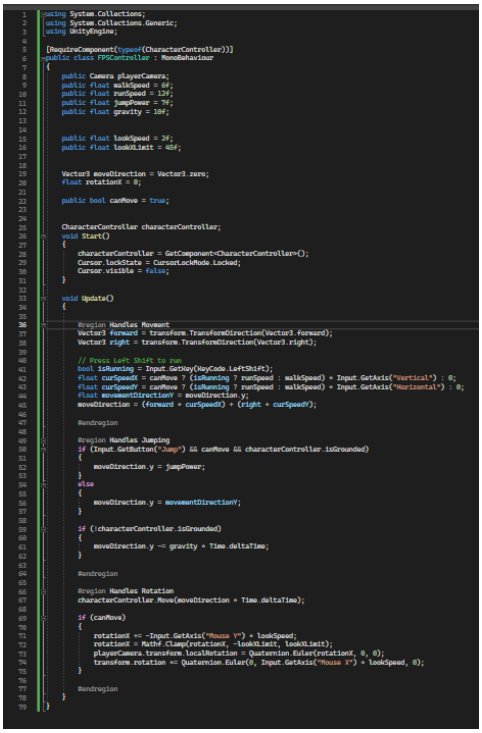
Test ID	Objective WHY	Prototype & Test Description WHAT	Results to be Collected HOW	Estimated Time WHEN
2	Ensure the player can move properly	Set a list of paths meant to correspond to certain inputs, and run these using a “movement” script in an undeveloped environment.	Mostly qualitative observations 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
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	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.

		using paths and inputs, not also causing collisions between the player and walls to see how they interact.		

Prototype Description

To complete this prototype, a public class was used to facilitate creating a code that would allow the character to rotate and move based on cursor movement and certain keys. The user is represented by an empty game object that can move in the virtual space, as shown to the right. Once the code is written, it is added to the player as a script which can be updated after each test.

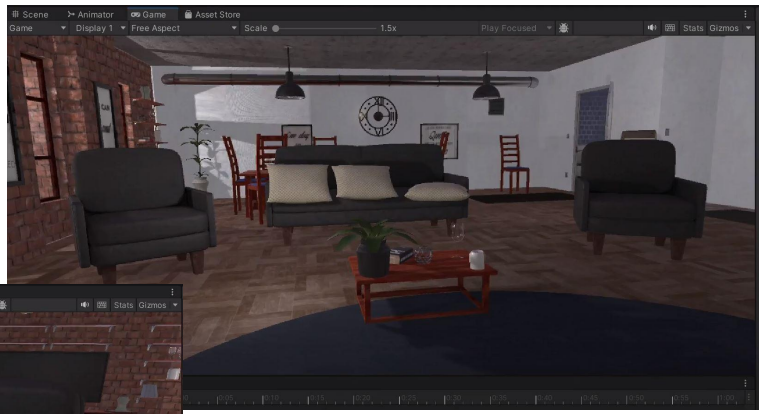
Image of Code

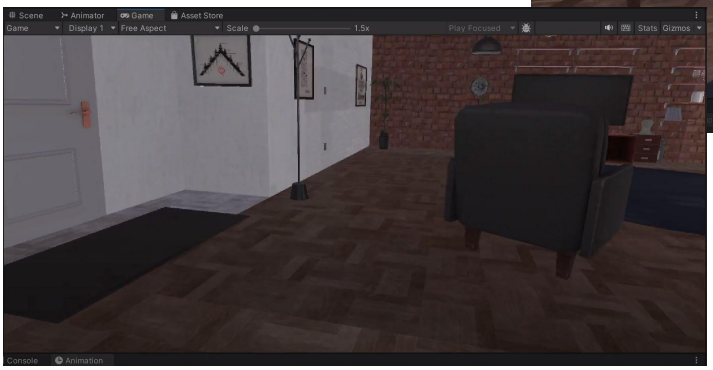


```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

[RequireComponent(typeof(CharacterController))]
public class FPSController : MonoBehaviour
{
    public Camera playerCamera;
    public float walkSpeed = 6f;
    public float runSpeed = 12f;
    public float jumpPower = 7f;
    public float gravity = 10f;


    public float lookSpeed = 2f;
    public float lookXLimit = 45f;


    Vector3 moveDirection = Vector3.zero;
    float rotationX = 0;

    public bool canMove = true;


    CharacterController characterController;
    void Start()
    {
        characterController = GetComponent<CharacterController>();
        Cursor.lockState = CursorLockMode.Locked;
        Cursor.visible = false;
    }

    void Update()
    {

        #region Handles Movement
        Vector3 forward = transform.TransformDirection(Vector3.forward);
        Vector3 right = transform.TransformDirection(Vector3.right);

        // Press Left Shift to run
        bool isRunning = Input.GetKey(KeyCode.LeftShift);
        float curSpeedX = canMove ? (isRunning ? runSpeed : walkSpeed) * Input.GetAxis("Vertical") : 0;
        float curSpeedY = canMove ? (isRunning ? runSpeed : walkSpeed) * Input.GetAxis("Horizontal") : 0;
        float movementDirectionY = moveDirection.y;
        moveDirection = (forward * curSpeedX) + (right * curSpeedY);

        #endregion

        #region Handles Jumping
        if (Input.GetButton("Jump") && canMove && characterController.isGrounded)
        {
            moveDirection.y = jumpPower;
        }
        else
        {
            moveDirection.y = movementDirectionY;
        }

        if (!characterController.isGrounded)
        {
            moveDirection.y -= gravity * Time.deltaTime;
        }

        #endregion

        #region Handles Rotation
        characterController.Move(moveDirection * Time.deltaTime);

        if (canMove)
        {
            rotationX += -Input.GetAxis("Mouse Y") * lookSpeed;
            rotationX = Mathf.Clamp(rotationX, -lookXLimit, lookXLimit);
            playerCamera.transform.localRotation = Quaternion.Euler(rotationX, 0, 0);
            transform.rotation *= Quaternion.Euler(0, Input.GetAxis("Mouse X") * lookSpeed, 0);
        }

        #endregion
    }
}
```

The code starts by initializing the user speed, making sure that the speeds are such that there are smooth transitions between different directions and movement. Changes in character position in the x or y direction are controlled, through if-else loops by the state of one of the components accessible in Unity (characterController) or the state of a boolean (canMove). The actual change in movement/direction is controlled by various variables, taking into account direction (by WASD input handling), cursor speed, axis of movement, and position in vector space.

Images of House Asset





Video Demo of Movement Code in Action

Design Project - Prototype 2

Model Used For Tests

We used an experimental model for this test. An experimental model was the best option for this test because it was the most effective and simple model for our case. It would have been ineffective to make a numerical or analytical model of the house and movement code when we could instead use the Unity software to make it happen and test it experimentally. Our model is fairly high fidelity because we made it in the same software as our final product will be. This makes the results of our test more valuable because the movement and house model closely resembles our vision for our final product.

Test Results & Observations

Once we completed our prototype and the model all we had left to do was follow the test plan to see if the prototype was successful. We decided that we only needed to run two tests to check on our three Test ID's as two of our Test IDs could be tested at once (Test ID 2 & 5).

Specifics of Tests

1. The first test we ran was a simple observation based test to determine if the asset we purchased is realistic and functional.
2. The second test we ran was a movement based test to determine if the user could move in all 6 directions (right, left, forward, back, up and down) all without leaving the confines of the home. This test was run 3 times.

Test 1 Qualitative Observations

Areas where the prototype worked as intended

- House looks like a traditional house
- Windows are functional IE can be looked through
- User could still move in the house with the assets imported
- House looked realistic and not overly animated

Areas that need improvement

- Couch cushions seemed to be floating over the couch
- Template items that need to be customized

Test 2 Qualitative Observations

Areas where the prototype worked as intended

- User was easily able to move around the scene with the WASD keys in all 6 directions

- Movement was smooth and intuitive
- User was able to control their speed of movement as to stay in control

Areas that need improvement

- User was free to fly out of the house as the house does not confine the player
- User was free to move too far in the upwards direction which takes away from our intended level of realism
- User could move through household objects like chairs and tables

User Feedback

We decided to have our roommates and friends test our prototype as they likely have about as much experience with unity movement codes as the politicians 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 seemed unfazed by the layout of the house and did not make any remarks about the house which suggests that our simple and traditional background is working as intended
- User 1 quickly figured out how to move around the scene and explored the layout of the house
- User 1 initially seemed shocked when they were able run straight through elements of the house

User 1 Quote: "That chair is only for show"

Feedback of User 2

Qualitative observations of User 2's experience

- User 2 took roughly 10 seconds to look around the house with the mouse before asking "What do I do?"
- User 2 has no experience in gaming so they did not understand that they needed to press WASD to move.
- User 2 continued to look around but never figured out how to move. User 2 remarked "Very pretty house"

User 2 Quote: "How do I move"

Feedback of User 3

Qualitative observations of User 3's experience

- User 3 looked around the house for around 5 seconds before moving no remarks about the house layout or style

- User 3 originally attempted to move with the arrow keys before trying WASD and succeeding
- User 3 moved around the house on the ground level for around a minute, exploring the house before they realized that they could fly and were not confined by the house and flew through the ceiling.

User 3 Quote: “What happens if it rains?”

Updated Test Plan & Next Steps

At this stage we have completed two prototypes and a test (Test ID 2,3,4 & 5, in green below). 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 fleshing out the house so that it shows the effects of killer robots and integrating our interactivity to the house (Test ID 6 and higher fidelity prototype of Test ID 4, shown in blue). We decided that Test ID 1 was sufficiently covered throughout the past two client meetings and would not add to the project as much as the other prototypes/tests (shown as a strikethrough text).

Overview of Prototypes and Tests

Test ID	Objective WHY	Prototype & Test Description WHAT	Results to be Collected HOW	Estimated Time WHEN
~~1~~	~~Ensure that the storyboard communicates our main message~~	~~Make a basic storyboard which has screenshots of the assets we are planning to use and communicates the intended game play and scene structure~~	~~Qualitative observations based on other students, and hopefully the client, of how well the story board communicates risks of autonomous killer robots~~	~~About one hour to make the storyboard and a 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 they interact.	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.
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	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

		windows, locked doors, etc.). Some lighting changes would make the outdoor setting more realistic		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

Our progress in this second prototype has brought us closer to our envisioned product. We focussed on movement in this iteration of our VR simulation whilst incorporating our chosen house asset. Feedback from users and the client has identified our successes in making this project realistic to current research on the impacts of autonomous weapons and accessible to as many users as possible. It has also shown some areas of improvement in making sure that the movement and interaction of each part/scene of the VR is easy to understand or figure out, making sure that we have enough time to communicate the main messages of dehumanization through usage of autonomous weapons. Completing the rest of the test plan and iterating it as needed will ensure that our final project meets the client needs, crafting an immersive simulation of the societal implications tied to autonomous killer robots.

Wrike Link

https://www.wrike.com/frontend/ganttchart/index.html?snapshotId=OBPqQoyW8ThbTBUeUaicLJF5oe6BCcKA%7CIE2DSNZVHA2DELSTGIYA