Homework 2 Project

Virtual Reality Game

Class: CS/IDEA 310H (Mixed Reality Design)
Assignment: Homework Assignment 2
Year: Fall 2024
Members: Christine Surber

Table of Contents

Abstract

This assignment tasked all the students to create a game, once again, but instead in the form of virtual reality. The goal was to create a game from scratch (while still able to use outside assets and templates) to fill in all the required features.

The game needed to have:

- 2 levels
  - 1st level is gameplay
  - 2nd level is winner environment
- Teleportation area for players to traverse
- Unlockable area
- Modified pistol blueprint
- At least 3 unique grab interactables
- At least 5 targets in the level
  - Tracker to count how many have been destroyed
- Spatial UI

Extra Credit:

- VR Menu Scene level
  - Play & Quit button that function as they say
- Moving targets
- “secret” room that is unlocked by a “key”

All of this adds up to 150 points (not including the extra credit). The goal is to give you starting materials and guides and let you take the initiative to figure out the best ways to make the requirements happen.

This document will describe all the parts and details that it took to build the game and what went into it. Everything from the resources, how the game works, to the assets used to build the game.

Resources

1. YouTube Videos
2. ChatGPT
3. Class Lab Videos
4. Previous Projects
5. Outside Assets
6. Unreal Engine Assets
7. FAB (UE Marketplace)
8. Game Design Guides

YouTube Videos

Setting up VR

- How to Setup VR in Unreal Engine 5.1
- UE5 VR SMOOTH LOCOMOTION | Unreal Engine 5.2 Tutorial + Environment Showcase
- Smooth Moving [Locomotion] & Smooth Turning | Unreal Engine 5 [VR] Tutorial
- How to Make a Moving Platform in Unreal Engine 5
- How to Change Levels with Fading In Unreal Engine 5

UI Design

- Make A Professional MainMenu And A Settings Menu In Unreal Engine 5
- Build User Interfaces In VIRTUAL REALITY For Unreal Engine 5

Learning

- Set Timer By Function Name/Event | Unreal Engine 5 Tutorial
- How to work with the Timeline node in Unreal Engine 5

CSU Videos

- CS310H (Video 10) Fall 2024 Created By Brendan Kelley. VR VIdeo 2
- IDEA CS 310H Lab11 VR 3IDEA CS 310H Lab11 VR 3
- IDEA CS 310H Lab12 VR 4

ChatGPT

I would often go to ChatGPT for:

- Explanations on how to do something
- Explanations on what my errors mean in Unreal Engine Blueprints
- Finding videos on how to do something
- Processes on how to perform a task in Unreal Engine

When I first started using ChatGPT it was originally to help explain to me how things worked or why a process I was trying to implement wasn't working. I tried to use it to find videos on YouTube for me to save me time from scrolling and trying to find the video that worked for what I needed. In this case, on this project, I used ChatGPT much more for explaining processes to me. I would give it a task and through explanation of what I wanted it to do and ask it to walk me through how to make it work. I used this multiple times for implementing the Targets functionality.

Outside Assets

The only outside resources that I used was the FAB lab that is attached to Epic Games content. Other than that, I didn't really use anything else.

Unreal Engine Assets

I used a couple different Unreal Engine assets to build my game. They were either already attached to Unreal Engine or I got them from the FAB Labs.

Assets I Used:

- Modular SciFi Season 1 Starter Bundle
- Stylized Nature Pack
- SimpleDesign_TableGroup01
- UE Starter Content

Level Design

- Level 0 – Main Menu – Modular SciFi_COMM_EX1
- Level 1 – Exploration – Modular SciFi_INT_EX3
- Level 2 – Winner – Stylized Nature Pack Assets & Static Meshes

Game Mechanics – How It Works

And now for the detailed part of how the game works…

Game RunThrough

Level 0 – Main Menu

The level begins with the player in a cyberpunk/sci-fi themed containment center with a center consol and a large window that looks out upon a galaxy with blues, pinks, purples, and whites. There is a table behind the player that displays robotic round cats, some candles, chairs, and a table. There are also columns with fire in them on each side of the room. The menu is in the right corner that the player can interact with to start the game or exit. The columns act as a secret lock so that the player can place one of the special cats with pink ears into to unlock the door to go outside and get a better view of the skybox that surrounds the center. The player can go outside and then start the game to the exploration, activity-based gameplay to progress.

Level 1 – Gameplay Level

The player is teleported to a different part of the containment (another level) where they are greeted with a small room with two columns that have robotic cats inside of them. This hints to the player that maybe that's where they should go. The door to the next room opens where the player can grab pistols, candles, chairs, and robotic cats. The player has two options: the regular door or the secret door. The secret door is unlocked by the pink-eared cat being placed in one of the columns. The regular door is unlocked by the player shooting all the targets in the room twice. Once the target is hit twice then it will be destroyed and a counter will increment. Once all 3 targets have been destroyed then the regular door will be unlocked so the player can move onto the next part of the level. The player will have to shoot 3 moving targets this time. Each target moves at it's own speed but the same concept applies. Once the player shoots all 3 targets then the first door will unlock but the second one doesn't. The player must put one of their guns in the column next to them to pay the Gun tax. This will unlock the last door which will show the player a doorframe with signs that tell the player to walk through it. Once the player walks through the frame then they are transitioned to the next level.

Level 2 – Winner Level

The player arrives in a simple world with clear skies and a couple trees and rocks surrounding them. There is a sign in front of the player with a chest that erupts sparks out of it saying that the player won. There is a reference to the Robotic cats sitting next to the winning sign. The world is chill and simple to represent that the player made it through the space adventure and made it back to land on a planet.

How Each Asset Works

Player Character

The player character is based on the default VR character with just the hands being visible to the player. I thought about trying to implement the ability to see all of the character with an FPS view of the *ThirdPersonCharacter* Mannequin but decided that that would take too much time and wasn't necessary. The assignment asked us to use the default *Teleportation* method of movement but decided I would change that into *Smooth Locomotion*. I wanted to be able to move around with the controller thumbsticks and my head. I also wanted to do this because the teleportation kind of messed with my sense of balance and made me feel slightly dizzy. I'm so used to playing games like VRChat (with Full-body Tracking) so I wanted to keep that the same.

Targets

For the targets, I decided to keep them simple and use a static mesh of a pillar but form it like a target used for archery or guns. I didn't have enough time to make an actual target in Blender so I just improvised with what I found on the FAB store. I created a main Targets blueprint to keep track of the important functionality but also created children of that blueprint to help keep track of different parts of the level and counter. Everything is organized by color and with comments to make it obvious what is happening and why. The targets have a Text-render above to display the hit count of the target. If the target is shot once then it will change from a concrete material to a copper material. If the target is shot twice then it will change to a rust color then explode and be destroyed. This will then update the counter.

I created a child of this class and added a few extra pieces to it so that it could be a moving target. I added the *InterpToMovement* Component so I could set it to move from point position to point position. I added a target speed variable so I could control how fast it moved between points. There is also a MoveToPosition variable that is instance editable so the player can choose where they want the points to move back and forth between.

DestroyedObjectsTracker

This is a blueprint specifically designed to track and count how many targets have been destroyed. Whenever a target is destroyed a counter will increment and display on a text render that is connected to the counter. By creating an event and adding to it in the *Targets* blueprint we are able to keep track of how many targets have been destroyed in the world and for other aspects of the game as well.

This blueprint also unlocks the next part of the level in the beginning (no-teleport-area). Once the tracker reaches 3 or more then the wall to the next part of the level will be destroyed to allow the player to move onto the next part of the level. This will open up the player to continue with the obvious path of the level and continue the rest or find all the pieces to unlock the "secret" door.

SecretDoorWall

This blueprint is designed as a wall that can be unlocked or destroyed based on the number of targets that have been destroyed in the level! The static mesh is a wall of some kind and has an event called "DestroyWall" which says to destroy the wall. This is connected to the *DestroyedObjectsTracker*. If the tracker reaches 3 or more destroyed targets, then the first wall in the level will be destroyed and open up the next part of the level. Throughout the level the player must keep destroying targets (each progressively getting harder and harder) to unlock all parts of the level.

Once the player unlocks all the parts of the level then they will be given an option to take the obvious route to finish the level or the secret route to finish the level! This secret door will be unlocked by a key (*KittyKey*).

RoboRoundCat & KittyKey

RoboRoundCat is the parent class that will keep the basic functionality for the child classes. *KittyKey* is the key to unlock the secret door. The cat will be placed on the *UnlockableColumn* which will then unlock the secret door. Both cats are grabbable so the player can grab the object and throw, hold, or place the object wherever they want.

UnlockableColumn

UnlockableColumn is the parent class to the children of this actor. I decided to make children based off it to organize the game and keep track of what is important and what isn't. The parent holds the basic functionality like the other blueprints and then the child will hold the important functionality specific to that class.

The column is made up of a static mesh comprised of a pillar and a pillar bowl that objects can be placed into. Inside the bowl there is a flame particle effect and a box collision which will be used as an overlap for the key to unlock the secret door.

AmmoPistol

I took the default pistol of the game that came with the VRTemplate and adjusted it so it had a text render to keep track of ammo and a reload function. The weapon can be reloaded by the controller and can tell which hand the pistol is being held in so it doesn't reload both guns if more than one is being held. It has sound effects to indicate that the gun is being reloaded as well.

UnlockableDoor & Other Doors

This is pretty much the same as the Secret door files just slightly different animations.

Many Signs (Text)

I created a few blueprints where it is just static mesh(s) and text render to communicate to the player. Basically, it just signs that the player can read to understand what is going on in the level.

Candle

Static mesh with a flame particle effect to make it look more realistic. It has a grab component but only works on the candle without the particle effect.

NextLevelDoor

This is a blueprint of a static mesh and a box collision. If the player collides with the box collision then it will transition into the next level.

Development Process

The second Homework game project went much smoother than the first project. I still had limited time to make the game work but learned from the first project and designed a more efficient way to develop the game in a timely manner. The biggest difference here is that I created an excel file to keep track of all the important parts of the game that I need to have implemented. I based it on percentage to give me a better idea of how far I am into the game. I had a better understanding of how Unreal Engine worked and how to connect different parts of the game.

Instead of spending all my time on visual appeal I spent majority of my time working on the functionality of the game and then working on visual appeal. As long as the game works as it's supposed to then I have done everything correctly. My biggest challenge was trying to make the extra credit main menu level work. I was struggling to get the UI widget to be interactable with the player. This was the biggest stressor of the project and took up lots of time. I was finally able to make it work after asking ChatGPT and finding the right video to explain the process (and it was quite a process).

It took me on average about 30 minutes to implement each part of the game which is much quicker than Homework 1. I was able to pace myself and use my previous experience from the last project to better construct the game and utilize assets.

Challenges & Solutions

My biggest challenge was the Main Menu! It took me hours to figure out how to make an interactable menu work in VR. The menu worked just fine in PC mode, but didn't work in VR mode. I tried watching videos, asking ChatGPT, and tried to look at the included code that came with the VRTemplate. After many hours looking back and forth at different resources I finally had the idea of looking at the already made menu that appears on the players wrist when they press a button on the controller to bring up the menu. I duplicated the menu so I could test out different things and see what works. After multiple different tests I finally was able to fix up something that worked. I unattached pretty much everything from the menu except the UI interface input and the connection between player hand and the keyboard and mouse buttons. I also changed up my UI Widget to be less complicated and changed some of the settings like the color of the hover option. Somehow, along the way of testing I changed something that ended up working. I'm not complaining, but I'm just glad that it finally worked.

Another issue that I came across was the issue of trying to find the best way to communicate with ChatGPT. I wasn't always able to word what I wanted in enough detail so it would skip out important parts of instructions that I didn't know or wasn't able to fill in the gaps because I'm still in the learning process of understanding Unreal Engine. I got better overtime and also knew what kind of videos I wanted to look for. There was also the issue of having trouble finding the right videos that went over VR implementation in Unreal Engine 5. There weren't as many as PC videos, which makes sense, but was still frustrating.

Conclusion

In conclusion, this project was much more efficient, higher quality, and all-in-all really fun. I learned from my last project and was able to use my time better and create a higher-quality game that feels like you're actually moving through a level. I feel really happy with how it turned out and I didn't even have to stay up until 1 or 2 am.

I was able to learn how to communicate with ChatGPT, search for videos, and develop a better understanding of Unreal Engine in general. This game was also made in less than a week due to school and all the other projects that I had to work on. But again, I learned from last time and was able to use that experience to my advantage and set up a work schedule that worked with the due date of Monday.

I am very proud of how this game came out and I hope that all who watch or play it also think that it's pretty cool. I had support from my mom and boyfriend who constantly told me that it'll turn out amazing no matter what I came up with. I feel like I was able to be a lot more creative with this game than the last game and it only makes me want to make another one in my spare time. I look forward to future projects and everything else that I can learn and contribute to.