VR Project

Nahfila Ali

I. Introduction

When I initially started my project, I wanted to create a park/museum since I couldn't pick between the two. I wanted to work with nature assets, and also building assets so I could gain experience in both. I also thought about creating animals in nature, weather animations, and character interactions. This ended up being very ambitious since I spent a lot of time figuring out the tools in Unity (the software I used to create my final project). My main objectives were to put down an interesting terrain, animate a character, create some sort of interaction, and use several assets. Through this, I would have learned the basics of Unity and creating a VR application and would be able to apply it to much more complex projects in the future.

II. Terrain

One of my favourite things to work on in this project was the terrain. My initial demo video showing my project was just the terrain I created. I used a YouTube tutorial that showed me how to download terrain tools, which also showed me how assets and tools can be downloaded into Unity projects. I particularly enjoyed sculpting out the area and creating neighbouring terrains, and I experimented by creating different environments, such as snowy mountain or shallow rivers, and I ended up deciding on giving my project a green field since I would also be able to use the dirt paint to make the terrain look more realistic. I was surprised about how realistic the terrains looked, especially after painting them. While painting the terrains, I also learnt how to change prefabs, and I learnt that the actual paint you use to draw grass or dirt onto the terrain contains three layers. It seems that one layer is the main colour and the other two layers help create shadows or highlights. Overall I enjoyed creating the terrain and it ended up being the most exciting part to witness on deployment.

III. Asset Collection

Downloading assets was an interesting part of the project, as it reminded me of other games that use the same features, such as Sims 4, where a player is able to download crowd-sourced assets in a similar manner for their game. I didn't find this part to be particularly engaging, but it definitely made my project look better than if I had just kept the terrain. When looking for assets, I expected there to be several websites dedicated to creating Unity assets (because of how popular the Unity software is), because this has been my experience when dealing with assets downloading for other software/games. I found, however, that most assets are directly found in the Unity asset store instead of any third-party website. I considered attempting to create my own asset, but I soon realized why many assets are behind a paywall, as it seems to take a lot of time and dedication to not only build the asset but also consider the material you use and the physics around building said asset. One thing I realized about assets is that they take a lot of space and can cause latency issues within the program, and so far the only solution I've found is scaling down the distance from which you can see the assets (seems to use less memory and creates a smoother environment). While this wasn't an issue for my project, as I only had a couple of assets, I can easily see this becoming an issue for a much bigger project.

While I was downloading assets, I added some walls to my project that went along a path that I created in the terrain, and put some pictures on it. This was slightly difficult because the downloaded assets and the Unity cube I sued to make the wall didn't want to work together, and I had trouble with the painting going through the wall, or not being flush to the wall. In the future I would like I figure out a smoother technique to create objects like this where I would need to match axis-es.

IV. Animation

My favourite part of this project was exploring animations within Unity, and this is where I learnt the most. My first experience with Unity animations was for the demo I created showing how to download animations from Mixamo and use them in Unity. I really ended up enjoying Mixamo, and the animations were much more fun to explore than the other asset collections. While doing research for the demo, I realized that just like the static assets, creating animations was very difficult, which is why there were so many pre-made ones already. It seems that to create a humanoid animation, you would have to create a humanoid object that has its own joints, feet, legs, ligaments etc. I was able to download a humanoid sprite from Mixamo and see exactly how each part of the body moved in a tab called ”Animation”, which let you see the curves, the run time, and the specificities of a single animation. I didn’t end up using that feature for much more than my own interest, but it is something definitely want to explore in the future if I consider making my own animation. While I couldn’t create a humanoid animation, I also found that Unity let’s the user create their own animations with simple 3D objects, such as cubes or spheres. I experimented with this by creating animations where an object moved form one side of the terrain to the other. I found the ’Animator’ tab with the animation state machine to be the most compelling part of animating because of how intuitive and useful it was. I particularly appreciated the ability to trigger events using a ’Boolean’ or a ’trigger’ parameter in during transitions. In my final project, I paired the animation state machine with my code to create interaction where the player could knock over a character.

V. Coding Portion

My least favourite part of the project was the coding portion, and unfortunately this is where I spent a majority of my time. At first I had assumed that this would be simple, especially because of the amount of resources available online, but I soon realized that I would have to figure out the logic on my own and I would only because able to use some functions from the Unity provided documentation. My plan initially was to write code that would trigger and animation for my character on impact with the camera. In other words, the player would be able to knock over a dancing humanoid sprite simply by pushing it. However I wasn’t able to do that, and I simply ended up writing code that I attached to my sprite. This code would let the player tap the humanoid, which would trigger a Boolean variable ”isPushed” to true, which would tell the sprite (within the animation state machine) to switch to a falling down animation. In the code, the variable would reset to false, and the character would get back up and start dancing again until it was pushed again. This seemed like a simple idea to me at first, but I realized that to create an interaction, I would have to consider every part of the objects, such as the animation, the physics, the material and the code. Something I ended up learning was that Unity has default physics. I used the Mesh Collider to help my objects detect whether or not they were pushed/tapped, and I initially used RigidBody to make my objects adhere to Unity physics, though this ended up being useless when I decided to just change animations by tapping on my character.

VI. Camera Motion

A large portion of my code was dedicated to figuring out how I could move the camera by just tapping on the screen. For my project, I used Google Cardboard, which limited the user input to only tapping on the screen. So to move through the terrain, I wanted to write code that would let the use move in whichever direction they tapped. I wasn’t able to achieve this exactly, but I was able to write code that would let the player swipe in the direction they wanted the camera to move. This code was by far the most complicated code because I wasn’t able to find many videos that were unique to my situation. I ended up using the Unity functions documentation a lot, and observed how the code was being used to piece together my logic. I ended up using similar logic when I created my character interaction. Since my screen was already able to detect touch, I decided that it wold be easy to implement the same function and attach it to my sprite. Something I wasn’t able to implement in the code was moving the camera across the y-axis, which had the unfortunate effect of periodically clipping my camera through the terrain if the terrain was higher than the camera (for example, if I maneuvered my camera towards a mountain, the payer would just see through the terrain instead of moving on top of the terrain). In other words, my camera moved along the x-axis, but ti did not adhere to the terrain. This could possibly be because the camera didn't recognize the terrain as a rigid body, but for the functionality of my project, it wasn't important, though I would like to improve on this in the future.

VII. Brushes

I briefly used brushes while adding shrubbery and trees to my project, and I found that this was much better than adding several tree assets. The ”tree” brush didn't interfere with how the project performed. I think the most interesting part about the brushes was that I was able to download anything from the unity asset store and turn it into a brush, making it easier to lay down. For example, fences, bushes, flowers etc. The only problem I had was that the player is unable to see the trees from a far distance, since the painted objects seem to differ from assets or other 3D objects (which you can see from a distance). The tree or plants would become more visible as the player walked closer and walked through, possibly to prevent latency issues. This blunder ended up giving me some insight into how some video games I've played have worked, as I've seen this same phenomenon in them.

VIII. Future Endeavours

While doing this project, I learnt a lot about the basics of creating a Unity project beyond just downloading assets. I was able to learn technical skills like animation, character creation, coding, etc. In the future, I would like to create a much more complex project. Most of the time I spent on my project had been reading through documentation, debugging, and learning, so the next time I pick up Unity, I'll be much more familiar with the learning curve. Specifically I would've liked to create weather animations for my project, which is something I didn't have the chance to explore (though the terrain did have weather features such as water and wind erosion that would help shape your terrain). Something else I'd like to tackle is the latency issue that seems to come with too many assets within a single project.

Conclusion

To summarize, I enjoyed learning the basics of the Unity engine, like animation, brushes, interaction code, terrain techniques, and assets, but I would like to use these skills to create a much more complex game in the future, or possibly build on the project I already have.

References

[1] S. Tutor, “Playing animation on trigger in unity,” Online video, 2019, https://www.youtube.com/watch?v=JS4k_lwmZHk.

[2] D. Krossing, “14 — collisions and triggers in unity — unity for beginners — unity tutorial,” Online video, 2022, https://www.youtube.com/watch?v=MfKyUkZb1V4.

[3] Brackeys, “How to make terrain in unity!” Online video, 2019, https://www.youtube.com/watch?v=MWQv2Bagwgk.

[4] A. Fenwick, “Unity fence layout tutorial,” Online video, 2020, https://youtu.be/vT0lnwtPz3o.

[5] Unity, “Blend trees - unity official tutorials,” Online video, 2014, https://youtu.be/HeHvlEYpRbM.

[6] A. of the First Video, “Title of the first video,” Online video, Year, https://youtu.be/bZW2IEmwOEg.

[7] C. Monkey, “How to fix read-only animations (and edit humanoid animations inside unity),” Online video, 2023, https://youtu.be/HvpdSrGCePc.

[8] GDi4K, “Unity: How to build run on iphone (2020),” Online video, 2021, https://youtu.be/DzH4UODL7Dc.

[9] KaiJPR, “How to paint texture, add grass flowers to terrain in unity 2021. ep2,” Online video, 2022, https://youtu.be/jKseULf3z0c.

[10] Unity Technologies. (2022) Unity documentation. Unity Technologies. [Online]. Available: https://docs.unity3d.com/Manual/UnityManual.html

[11] nahfilaali, “Vrprojectunity,” Online, 2023, https://github.com/nahfilaali/VRPROJECTUnity/tree/main.

[12] N. Ali, “vr final - hd 720p.mov,” Online video, 2023, https://youtu.be/tpUaC5YK8bM.

[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]