**Personal Reflection  
Personal Portfolio 3 – VR**

Before I detail my process and my thoughts on it, I want to clarify that most of this project was worked on in Q1 of 2022-2023, so my retelling may be a bit hazy. What’s more is that the masterclasses were followed in Q3 of 2021-2022, so forgive me for having some issues recalling details. **Reflection on process**Starting work on this project, I had heard of the many technical difficulties come from working on different VR systems daily. I started by looking onto the different VR systems available at the XR-lab and ended up choosing to work with the Quest 2, due to how “easy” it is to work with and set up for game development.

I spent about 5 hours on my first day getting the environment to work, which roughly lines up with my prediction from my first learning goal.   
After setting up the device, I started to develop a simple prototype including simple locomotion and object interaction using Unity’s build in XR Interaction Toolkit package. For more information about this package see [here](https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.3/manual/index.html).

I had a lot of trouble wrapping my head around how to work with this system, as it uses the new input system as a major part of its architecture. The Input System is a package from Unity that overhauls the way inputs are detected. For more information on the Input System Package, see [here](https://docs.unity3d.com/Packages/com.unity.inputsystem@1.6/manual/index.html).  
As I got closer to wrapping up development for the simple demo, I couldn’t shake the feeling that I really disliked working with the XR Interaction Toolkit. In hindsight, I think it is simple very complex in nature and covers a lot of ground within Unity that I am (still) very unfamiliar with.

For the demo, I ended up creating a simple shooting gallery style demo, with some UI and a score feature. I had some fun putting it together but opted to scale the demo down wherever possible.

I ended up trying to incorporate as much of LO 3 into the demo scene, as I had invested way more time into the small demo then I anticipated. Additionally, I got caught in a concept I thought off and really wanted to free up some space in this project to pursue that concept.

This little concept I thought off was to rewrite the entire code base that handles device inputs and translates them into the Unity Engine from scratch. In hindsight, I think doing this was an amazing choice, though a bit irresponsible of me. Essentially, I wanted to rewrite the XR Interaction Toolkit from scratch, and obviously scale it down tremendously.  
Luckily, the XR Interaction Toolkit runs on the build-in UnityEngine.XR namespace.  
Knowing this, I figured I should be able to use this namespace to rebuild my own XR interaction system from the ground up.

So, I created a new project and started with creating device registering and simple locomotion. I must admit that some of the not so functional code caused me severe motion sickness. But I eventually got to a point where simple locomotion was working and inputs could be easily read from devices.

After this I started working on object interaction. This is where I ended up spending a lot of my time, trying to get grabbing, releasing, and throwing objects to function just right.  
I encountered a lot of issues during this process and ended up restarting and redoing the architecture for this twice. Even as I am writing this, I would love to tackle it all over again, as I’m sure I can do a better job than what I wrote for the final iteration.

The hardest part of this system was easily throwing objects for me, and the main reason was that I really wanted the rotational velocity to translate from the controller to the thrown object. Turns out Unity calculates these using radians per seconds, while the device uses Euler angles, which are measured in degrees. What’s more is that the device calculates this in “headset” space, while Unity uses world space. This all combined into a very messy translation between these measuring methods, and the result is still okay at best to me.

I ended up giving up on making a simple game using my custom framework because of the amount of time I had already invested into this project, and because I feared I wouldn’t learn much from going through with making a small game.

**Final product reflection**  
First, I want to reflect on my decision to disregard my third learning outcome and replace it by building my own XR interaction framework. I think this was a great decision, as it allowed me to learn much more, and allowed me to learn a lot more about a very technical topic. As a programmer, I think this was more interesting and educational compared to making a game using the build-in framework.

Next, I want to address the large number of technical issues for this project. These issues are mostly a result from not having my own VR set to work on and needing to rely on the XR-lab. I’m very grateful that I was able to lend a VR set to work on for this project, but it sure caused me many issues.

I’m very happy with how the first small VR game turned out. I had a very hard time getting used to the architecture and still think it’s very complicated, but it was very informative, and I think the demo turned out well.

Lastly, I am really glad I chose to pursue creating my own interaction framework. I learned a lot about architecture and device communication because of it. It has to be one of my favorite projects I got to work on during this study.

I also think that the two prototypes I made can be a nice piece on my portfolio, especially if I want to work for a company that creates interactive VR experiences.

**Reflection on Masterclasses  
Video Game Economics**In this masterclass I learned a lot about how to approach balancing, and how important economy is to the final play experience. It may not be a topic that is relevant to my everyday work, I still found it deeply fascinating and am very glad I attended this masterclass.

**Fast Prototyping in Unity**This masterclass was very helpful in my opinion. It didn’t necessarily teach me anything entirely new, it did show me how to use my tools more effectively. Very interesting masterclass, very happy I attended this masterclass in particular.

**Timesheet of hours spent**

|  |  |
| --- | --- |
| Task | Hours Spent |
| Setting up devices + other hardware shenanigans | 10 |
| Research device and XR architecture | 5 |
| Create a simple tech demo using the build in Interaction Toolkit. | 15 |
| Create a simple game using the build in Interaction Toolkit. | 15 |
| Building a custom XR interaction framework | 50 |

**Did you complete your learning goals?**I believe I completed my first 2 learning goals. I touched briefly on my 3rd learning goal, but believe that while I did not complete it, the alternative that I did pursue was much more valuable.

**What do you want to work on next term?**  
I recently wrapped up my procedural art project and would really like to investigate another procedural generation method I came across while doing research for that project.

**What do you need for year 3 / 4 and beyond?**At this point I feel pretty confident in my abilities and think I’m ready for year 3 and 4.  
As for beyond that, the main objective is experience, as I know very little about actual work in the industry. Additionally, I feel like my portfolio is a little weak, and would like to look into better ways of formatting my portfolio as an engineer, as well as what engineer portfolio conventions are.