OSGC Developer Journal

**Lead-up (June 7 – 13)**

Up until today, I dove into developing VR environments with Unity to see how the workflow feels and what kinds of tools are available. I used the latest beta release of the SteamVR plugin for Unity because my headset uses SteamVR for playing games and applications, and the stable release doesn’t work well with the latest version of Unity.

I attempted to learn the tools provided by the plugin (prefabs and scripts to use as components for Unity’s ECS) by exposure, but found myself puzzled by many issues too difficult to pinpoint the cause of. An example was my attempt at recreating the HoverButton prefab using the same components it used. Eventually, I pretty much had a copy of the prefab, just using a cube and empty game object I created, but it still did not function properly. I broke down and used the prefab instead, and attached my logic to the press event (resetting the scene). When I reset the scene, I was greeted with yet another strange error. My position in the scene was reset as expected, but now, I had another right hand offset by the position my hand was at before resetting the scene. Upon investigation, I found that the Player component is treated as a singleton, but when running the game, the player object was marked with DontDestroyOnLoad, which causes it to persist across scene loads. I suspect this is what caused the issue, but I reached the end of my work for the day before investigating it further. I did decide to read the Article for the SteamVR plugin’s Interaction system to achieve a bit more insight as to how the system works. <https://valvesoftware.github.io/steamvr_unity_plugin/articles/Interaction-System.html>

**June 14**

Today, I worked on mocking up UI designs for the experiment. Phil and I have been emailing back and forth trying to decide what sort of project would be ideal to work on for this internship. After many ideas were considered and dismissed, we decided it would be best to stick with a procedural learning experiment where subjects are to use a touchscreen UI where they are prompted to perform certain procedures previously taught to them. A control group of subjects are to learn by video only, and an experimental group will use a virtual learning environment. To see if the VLE was more effective at teaching the procedures, both groups will undergo timed tests for a subset of the procedures taught to them using a physical touchscreen implementation of the UI.

At first glance, it may seem that the users of the VLE are at a clear advantage over those watching training videos, and I believe that’s true. In this situation, it is overkill to use virtual reality to teach someone how to use a UI since the testing procedure could easily have been used to train the individual in the first place; however, for much more complicated matters like teaching someone how to repair the outside of a spacecraft, it is much more costly to create a real mockup of the spacecraft and train someone to repair it. In that situation, virtual reality would be a much better solution because it would simulate being out in space with a spacecraft, but the only costs would be the VR system (a one time cost) and the development of the assets and interactions. The purpose of this experiment is to prove that procedural tasks are more effectively taught in VR than through other digital media.

June 15

After revising my UI mockup for phil, I explored another game engine, Godot, as a possible alternative to Unity. It is an open source and up and coming engine that runs on OpenGL. It claims support for XR, and boasts a very small executable size for most projects (about 15MB).

It seems to have a lot of potential given the heavy emphasis on using a layout-based UI system. The class I took on GUIs at OIT was also adamant about using layouts as it is the industry standard way of making scalable and reactive UIs. Although Unity does not have this explicit focus on layout-based UIs, I believe there is still room to utilize that methodology.

I create a simple demo UI resembling the mockup I’ve created in Unity and I think it has the potential to be exactly what I need. The tools made for developing VR experiences in Unity seem to have far more support than those for Godot, so I’m still gravitating toward Unity. If I can find that the interaction with UI in VR is easy to implement and manage, then I may switch over, but I must do more experimenting to find out.