

Assignment 3 - Fall 2019

Writeup

<Kushal Singh> <ks2377@cornell.edu> <11/20/2019>

INFO 5340 / CS 5650: Virtual and Augmented Reality
Cornell Tech - Fall 2019

Links

Make sure your URLs below are clickable hyperlinks in your submitted PDF.

Assignment repo

Replace “<insert link>” with an active hyperlink to your GitHub repo

GitHub: <https://github.com/virtual-and-augmented-reality/assignment-3-2019-ks20>

Assignment videos

Replace “<insert link>” with active hyperlinks to your solution recordings. They should be hosted on your Cornell Google Drive account. Make sure to set the video privacy settings as viewable/downloadable by other Cornell Google accounts.

Part 1: https://drive.google.com/open?id=1_KPI5GuzXrO40iPmRtzvMwA3yK1_pBel

Part 2: <https://drive.google.com/open?id=18f4Aje2xrZhESM9iAUrxoi-5C0MRh8Z4>

Part 3: Final Five: <https://drive.google.com/open?id=18f4Aje2xrZhESM9iAUrxoi-5C0MRh8Z4>

Video Recordings:

<https://drive.google.com/open?id=147cpDJbTEKiwhIEB10ISN1N3JDRET9hN>

Note: The order and total number of parts are different between A2 and A3, modify if needed.

Work Summary

Overall, the biggest challenge(s) I faced pertained to getting my environment set up and being able to test, build, and deploy the application. This is because testing small changes (such as lightsaber offset values and glue gun offset values) were significantly harder to do, since you couldn't test the x, y, z rotation and position values in the Unity simulator, but rather deploy it onto the Oculus device each time.

For Part 1, I was able to implement Locomotion and Distance Grabbing seamlessly by grabbing the relevant game objects and scripts from the DistanceGrab and Locomotion scenes. The

biggest challenge for this part was baking the lighting settings, as this took nearly an hour for me.

For Part 2, I was able to attach the power module to the lightsaber by detecting whether the power module on the desk entered the lightsaber's collision sphere. Once it did, I activated the hidden power module and de-activated the power module that the user picked up.

I was able to implement the Glue Gun by attaching a Particle System as a child of the Glue Zone GameObject. Once the GlueGun was picked up, I set the Particle System to active which allowed the glue to flow. Finally, after the glue was applied to the power module, I was able to attach the quillon and, ultimately, activate the lightsaber. The biggest challenge for this part was getting the Glue Gun to fire the glue when the trigger was pressed. The reason this was happening was because I had attached an OVRGrabbable Script and a DistanceGrabbable Script to the GlueGun GameObject. Since DistanceGrabbable extends OVRGrabbable, there must have been an overriding error when I was calling `grabState = this.GetComponent<OVRGrabbable>()` that was causing this confusion. So, to solve this issue I only added a DistanceGrabbable script to the Glue Gun, in order to provide me with both of the grabbing functionalities (distance grab and regular grab).

Final Five

For my Final Five, I added lightsaber sound effects when the user turns the lightsaber on and off. I did this because I wanted to adhere to Nielsen's heuristics of Match Between System and the Real World and because I thought that adding these effects would make it more fun for the user experience.