EE 267 Virtual Reality: Lab 7 (optional)

Instructions

Unity will not be supported by the course staff in office hours or on Piazza. However, for students who wish to use Unity for their final course projects this lab provides a good "getting started" guide.

Getting started with Unity

Unity is a cross-platform game engine that supports 2D and 3D graphics, drag and drop functionality and scripting using C#. If you have any questions about Unity, the best place to refer to is the comprehensive official documentation. You can download and install the latest version of Unity Personal (free) here: https://store.unity.com/. The download may take a while, uncheck `Visual Studio` in `Choose Components` during installation to speed up the process.

Modeling in Unity

An asset is representation of any item that can be used in your game or project. An asset may be a 3D model, an audio file, an image, etc. created inside or outside Unity. Unity has an asset store, which is a library for free and commercial assets. Access the Asset store by clicking on `Window` → `Asset store`. Search for models or textures or anything you might need for your project, download and import them. For more information about assets, refer to the official documentation on Asset workflow. You would be able to find existing models on the asset store. However, if you want to create your own models from scratch, this detailed video tutorial (https://unity3d.com/learn/tutorials/topics/graphics/models-and-materials) is a good place to start.

Making an interactive experience in Unity

GameObjects are an important part of Unity. It literally refers to every object in your game. You can create a game object and attach prefabs from your assets to it. The properties of the objects can be changed by either using the inspector window or writing a C# script. This allows you to control the position of your objects and add movement and so on. To make the experience interactive, we should be able to control some of our GameObjects by using external controllers (like the VRduino). And we should be able to render it in manner suitable for the HMD.

Let's look at the following:

Reading data from the VRduino

The position and orientation information from VRduino can be read in Unity and used to control GameObjects. In the unity starter project for Fruit ninja, there is a C# script `ReadUSB` which reads the serial port output from VRduino (programmed to output position and orientation), as bytes and parses the information. This script must be attached to the object that we wish to control, in this case GameObject2 (katana).

Render stereo images with lens distortion for the HMD

To render images suitable for viewing on HMD, we use the Google cardboard asset that is attached to the first-person controller. This takes care of the lens distortion and the stereo rendering. The screen size has been set to Nexus 6 to make it compatible with our screen.

To run the Unity Starter project (Fruit Ninja), do the following:

1. Connect a VRduino (programmed to output position and orientation) to your computer to act as the katana.
2. Note the name of the COM port and edit ReadUSB.cs script to have the correct port.
3. You may connect an additional VRduino and modify ReadUSB2.cs to add head orientation too.
4. Connect the HMD.
5. Click play in Unity and move the game window to the extended display on the HMD.
6. You can now play Fruit Ninja!

Use the provided Unity starter project to base your project. You can look for detailed video tutorials and demonstrations for developing a game in Unity on YouTube. FusedVR is a great channel to start with and made by former EE267 students!

Good luck with your project!