**OVERVIEW**

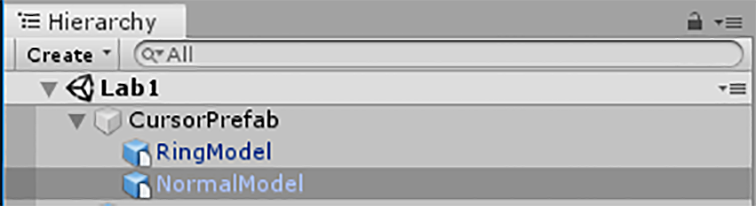
This lab is intended to get you started with VR development on Microsoft’s Mixed Reality Platform. This lab introduces the basics of Gaze interactions and how to make a standalone build.

**CLASS ASSIGNMENT**

* Setup the Odyssey with two controllers
* Demonstrate the Odyssey working
  + - * *You can do this by walking around the Mixed Reality Portal “home”*
      * *Show that you can use the controllers to teleport, and interact with objects.*
* You must use the PCs in Little 231 for this assignment
* **This is an in-class assignment that is due by the end of the lab.**

**LAB REQUIREMENTS**

* Create and Set up project for Windows Mixed Reality
  + Set Platform to UWP
  + Enable XR – Windows Mixed Reality in Settings
  + Setup Camera
* Import Package “Lab1Assets.unitypackage”
  + Use scene provided in package for this lab.
* Create a Cursor prefab from the package
  + Use assets provided in package
    - You have been provided with several Ring and Normal models choose one of each in order to build your Cursor Prefab
  + Structure your Cursor as following:
    - * Cursor *(create an empty game object) – this is the Parent Object*
        + Ring Model *(use an fbx provided) – a Child of Cursor*
        + Normal Model *(use an fbx provided)* – *a Child of Cursor*



* The Prefab Cursor Size should be <1.0f, 1.0f, 1.0f>.
* The children of Cursor parent should be scaled to make the Cursor look reasonable proportioned in the World. (See Gaze.cs in next bullet).
* Create a script named “Gaze.cs”
  + Place this script on the camera object.
  + **Start()** 
    - Instantiate the Cursor prefab
  + **Update()** 
    - Draw Debug.DrawRay() along ray cast line
  + Perform Raycast with a maximum distance of 50.0f;
  + On hit:
    - Set cursor object location to hit info location
    - Set cursor object rotation to hit info normal
  + On no hit:
    - Set cursor object location to 3.0f from the camera
    - Set cursor object rotation to point directly back at the camera
* Create a standalone build of the application
  + Refer to UnityWMRBuildProcess.docx for instructions on how to do this.
* Install and demonstrate the standalone build on a different machine on the date of or before the lab is due

**EXTRA CREDIT**

* Numeric ‘1’ key toggles rendering of a Game view visible/invisible “line”
  + “line” should be length of ray cast hit or 3.0f from the camera if not hit.
  + Make a cylinder a child of an empty object container.
    - Set size of the cylinder <0.1f, 1f, 0.1f>
    - Set the y size of the container to set the line length.
* ‘2’ key toggles normal portion of the model between shown and hidden.

**SUBMISSION**

Submit to Canvas a zip file with the following

* A folder that contains the standalone build of the application
  + *This is what you need to install the application*
* A folder that contains the Unity project
  + *This is what you need to launch the project from Unity*
* Remove files that can be rebuilt from the Unity project to reduce the size of the project folder
* Name you zipfile like this <YourName>\_AG231\_Lab1