Assignment #2
Quickstart to Creating Google Cardboard Apps with Unity

Released Date: Tuesday, January 12, 2016
Due Date: Tuesday January 19, 2016, 3:30pm ~ 5:00pm
Checkoff Location: Outside CSIL 4 in Crerar.

How to Complete Assignment: Yun Li will be available for assignment “checkoff” Tuesday 1/19, 3:30-5pm, outside CSIL 4 in Crerar. For Problem 2d and e, you will do a demo with the TA in the lab, and should turn in a 4-page writeup describing your system.

Checkoffs are individual, everyone should come with their own computer and mobile device, build the game using their computer and mobile device(s) and then show the game on the mobile device(s).

Please signup in advance for a 10-minute checkout slot, in this google doc https://docs.google.com/a/uchicago.edu/spreadsheets/d/1BOyb8--wQ8rR_3Q9fLMkKTVFDN_8qgY9DdqICf59ek/edit?usp=sharing

The purpose of this assignment is to expose you to 3D VR api’s for google cardboard. You will build a simple VR app where you can view a 3D object; the background changes depending on whether or not you are looking at that object. Then you will need to apply interesting changes to the Treasure Hunt Demo provided by Google.

1. Cardboard Development, Magic Lamp
   http://www.sitepoint.com/building-a-google-cardboard-vr-app-in-unity/
   Follow the above tutorial to understand how to develop an Google Cardboard VR app. This tutorial will guide you through the development of a simple VR game for Google Cardboard.

2. Treasure Hunt Demo
   After going through the Magic Lamp App, you will now need to go back to the Treasure Hunt Demo app provided by Google. You will need to apply changes to those games.

Assignment:
You will need to apply changes based on Treasure Hunt Demo.
   a. Change the cubic object used in the game. You need to use an asset from the asset store. I will refer that object as “Target Object” later.
   b. Add audio effects. Trigger audio when you are looking at (gazing at, pointer enter) the Target Object. You can find audio asset from Unity Asset Store.
   c. Add a button to the button panel. Every time that button is clicked will generate one more instance of the Target Object. Looking at (gazing at, pointer enter) the Target Object and click would destroy that Target Object.
d. Now we will introduce outside control over object state. You need to display the Target Objects as two different sizes in alternate 10-second intervals. So within each minute all of the objects should synchronously change color 6 times.

e. Now we need to add multiplayer interaction to the game. This can be achieved by the multiplayer support from Unity. Based on the game you have developed, you need to adapt it for multiplayer. First, you need to go through **Network and Multiplayer tutorial** ([http://docs.unity3d.com/Manual/UNet.html](http://docs.unity3d.com/Manual/UNet.html)). You will be able to learn basic concepts of developing an multiplayer game. Then you can start to develop your own Multiplayer VR game. The specific requirements are as follow:

i. You will need to use Target Object as your player object. You will need to see other player when they join the game (scene). You can do this based on the original Treasure Hunt Demo but instead of using cube as the player you need to use a more interesting object (such as the one used in a).

ii. When you (one player) are looking at another player (your VR cursor is on that player’s object), the appearance of that player should change, and that change should be noticeable to all other players. The appearance can change (color, size, etc.) as long as it is visually clear to other players.

iii. For this multiplayer VR app, you can see yourself (one player, an instance of the Target Object) and the other players.

Notes:

1. While developing the multiplayer app, you can use your mobile device and laptop to simulate a multi-player configuration. The laptop can run both server and client (with the laptop acting as a “host”), and your mobile device runs a client.

2. When you adapt the game to multiplayer, we suggest you not use “NetworkManagerHUD” GUI. Doing so causes problem because of the immature support of GUI in google cardboard. You can use NetworkManager API to enter different mode. If you do that, you can add more buttons to the button panel.

For elements a-c, a simple checkoff in the lab is sufficient. For d and e, you will do a demo with the TA in the lab, and should turn in a 4-page writeup describing your system (interface and code), and any salient design elements. Think about the performance of your system, and what kinds of interaction might be possible.