

## **Assignment #1:**

### **Quickstart to Creating Google Cardboard Apps with Unity**

**Released Date: Tuesday January 5, 2016**

**Due Date: Friday January 8, 2016, 1:30pm ~ 6:00pm**

**Checkoff Location: Outside CSIL 4 in Crerar.**

**How to Complete Assignment: Yun Li will be available for assignment “checkoff” Friday 1/8, 1:30-6:00pm, outside CSIL 4 in Crerar.**

**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 sign up immediately for 10-minute checkout slot -- here:**

**[https://docs.google.com/a/uchicago.edu/spreadsheets/d/1PbHuZdTEQTHrTq6uTz837WYxt\\_N8eoZs8QqQAZF\\_Ma8/edit?usp=sharing](https://docs.google.com/a/uchicago.edu/spreadsheets/d/1PbHuZdTEQTHrTq6uTz837WYxt_N8eoZs8QqQAZF_Ma8/edit?usp=sharing)**

Google provides two Cardboard SDKs, one for Android SDK and one for Unity SDK.

Android SDK is used for Android Studio. If we want to develop Google Cardboard apps with 3D models in Android Studio, we need to deal with OpenGL ES, this interface provides detailed control for power users. Given the short time frame for this class, we will focus on use of the higher level Unity interface for quick development of VR apps. However, you may choose to use this in your projects.

The Unity SDK can be used to develop VR apps for Android and iOS platforms<sup>1</sup> with Unity. For the class, we will use Android. Unity is a development platform for creating multi-platform 3D and 2D games and interactive experiences. It is designed to deal with 3D models directly and easily. In this class, Unity 5 is used as the development environment.

Unity provides two scripting languages, C# and JavaScript. Don't panic if you did not learn either of those languages. If you know Java then you should be able to handle C# for the two labs because they are very much alike.

Unity provides a wide variety of learning resources (<https://unity3d.com/learn>), including video tutorial, documentations, etc. You can use as much of these as you find helpful. However, here's a recommended learning outline.

#### **1. Install Software:**

Follow the steps at: <https://developers.google.com/cardboard/unity/>

After completing the above steps, you should have:

---

<sup>1</sup> We will not support iOS in 234/334 because of the cost of the SDK and the complexity of supporting multiple platforms.

- a. Installed Unity 5 Personal Version on your computer.
- b. Downloaded the Cardboard SDK for Unity.
- c. Downloaded the Cardboard Demo.
- d. Tried the Treasure Hunt Demo.

## 2. Work through the Roll A Ball Tutorial:

You will find the Roll a Ball tutorial at:  
<https://unity3d.com/learn/tutorials/projects/roll-ball-tutorial>

Follow this video tutorial and develop the app. The goal of the tutorial for you is to get familiar with the Unity IDE and app development procedures in Unity. While you may not understand how each component works while you are learning it, it is fine, just follow the steps. This is good practice for the use of rich libraries in development environments that you will not have time to fully understand.

*For additional background, you can find many more Unity tutorials at <https://unity3d.com/learn/tutorials>*

## 3. Work through Details of Unity:

Now you should dive into the details of Unity development. Unity provides video tutorials and documentations:

Video Material: <https://unity3d.com/learn/tutorials>, in the Topics section at the bottom of the website.

Documentations: <http://docs.unity3d.com/Manual/index.html>

You should complete the interface & Essentials, Scripting, Graphics, Physics, Audio, User Interface, Animation, Mobile & Touch materials. Its up to you if you prefer video or documentation approaches. This will involve range of experimentation and code changes. We encourage you to do more and explore.

## Assignment Elements for Checkoff:

- a. You need to change the Roll a Ball game so that it can be played on a mobile device. To be specific, you need to use acceleration information as the input on the mobile devices. To make it more tricky, you need to invert gravity in the game. For example, if you tilt the device, the ball should go up instead of going down.
- b. Change the color of every object in the game.
- c. Change the shape of the pick up objects.
- d. When you win the game, it should show a different message where your name should be included.
- e. If you'd like, you are welcome to add more changes to the game.

***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).***