CAMOSUN COLLEGE

COMPUTER SCIENCE DEPARTMENT

ICS 123 - GAMING AND GRAPHICS CONCEPTS

LAB 2 – CREATING AND LIGHTING A 3D SPACE

DUE: DEMO BY END OF CLASS MON JAN 30
You must work individually for this lab.
Submission Instructions are on the last page of the lab.

Objective:

To have the students create a rudimentary 3D space and then light it.

Preparation:

Supplemental Video Tutorials:

https://unity3d.com/learn/tutorials/topics/interface-essentials
https://unity3d.com/learn/tutorials/topics/graphics

Theory Part 1 – 3D Coordinate System:

For this lab, we are going to create a 3D space for a game. To accomplish this, we need to use a coordinate system to keep tract of the position of the player as well the positon of game objects in the space. In 2D, this is simple: you can use the Cartesian coordinate system (X and Y axes). X defines the horizontal axis; Y the defines the vertical axis. In 3D space, however, you will need a third axis – the Z axis. The Z axis is what adds the third dimension: *depth*. Every point in our 3D coordinate system will have three numbers – X (horizontal position), Y (vertical position), and Z (depth position), i.e. (X, Y, Z).

Unity uses the *left-hand coordinate system* for the Z axis. What this means is that the positive Z axis goes into the page. To understand this concept, hold up your left hand. Straighten your thumb out horizontally and point to your right. Straighten your index finger and point it up. Now straighten your middle finger and point it away from you. Your thumb is the X axis, your index finger is the Y axis, and your middle finger is the Z axis. If you imagine your thumb and index finger on a flat piece of paper, your middle finger (the Z axis) would be pointing 'into the page'.

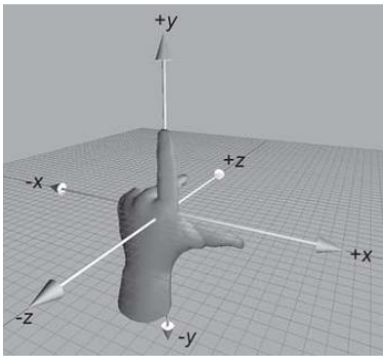

Figure 1: The left-hand coordinate system

Theory Part 2 – Lighting:

Once a 3D space has been created, you need to light it. There are three kinds of light you can add to your scene:

a. *Point Light* – Like a real-world lightbulb. Light rays emanate in all directions from a single point. Also like a lightbulb, the closer you are to the light, the brighter it will be.

b. *Spot Light* – Similar to point light but instead the light rays only emanate in a limited cone area. Usually used for specific highlighting.

c. *Directional Light* - Like the real-world sun. Light rays are parallel and emanate evenly across the scene. It is usually added automatically when you create a project since it is almost always used to light a scene.

For this lab, we will only use directional and spot lighting.

Tasks:

Tasks Part 1 – Creating the Scene:

1. Open up Unity and create a New Project (refer to Lab 1 if you don't remember how). Name your new project 'ICS123 Lab 2'.

2. Once the new empty project is loaded in Unity, go up to the *File* menu and select *Save Scene*. A scene in Unity is akin to a level in a game. Every game will have at least one scene and usually will have multiple scenes. For our labs, we will have a single scene. You can name this scene whatever you like! Once saved, the Scene will be added to the project's assets.

3. Now that the base empty scene has been created, you need to fill it with objects. Go up to the *GameObject* menu at the top and select *3D Object -> Cube*. This will create a new Cube object to place in the Scene.

4. This first object is going to be the floor of your scene. In the Inspector Tab, change the name of this object to Floor. In the Transform component of the Inspector Tab, set the Y Position to -0.5, the X Scale Factor to 50, and the Z Scale Factor to 50 (you'll need to zoom out in the Scene Tab to see the new Floor). You'll notice some extra components automatically included with a new Cube object (Mesh Filter, Box Collider, and Mesh Renderer). Don't change any of these components and you can ignore them for now. Your Inspector Tab should look like Figures 2 below and your Scene Tab should have an isometric view of a floor.

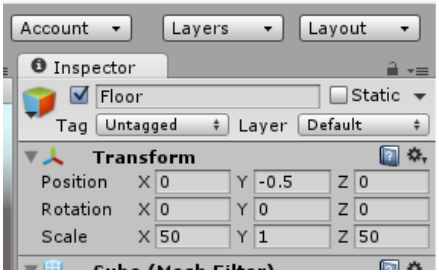

Figure 2: Transform Component of the Floor 'cube' in the Inspector Tab

5. Now create walls so that your Scene matches something like that of Figure 3 below. It doesn't have to match it exactly but it should have the same number of inner walls and definitely have the outer walls surrounding the floor. You shouldn't have any cubes going through each other. Start with the outer walls. Add more Cube objects to build the other walls. Besides entering numbers in the Inspector Tab, remember that you can use the Transform Tools (the buttons above the Hierarchy Tab) that you learned in lab 1.
   Here are some other hints:
   - You may find it easier to select objects in the *Hierarchy Tab* then trying to select them in the *Scene Tab*.
   - You can copy and paste objects with the usual shortcuts (Ctrl + c to copy and Ctrl + v to paste or right-click on the object in the Hierarchical Tab).
   - Double left-clicking on an object in the Hierarchical Tab will focus the view in the Scene Tab on that object.
   - Make your walls all the same thickness and height. The units used in Unity can represent anything you want. For example, in this case, it could be feet or metres. In Figure 3, the walls are one-metre-thick and four metres high!

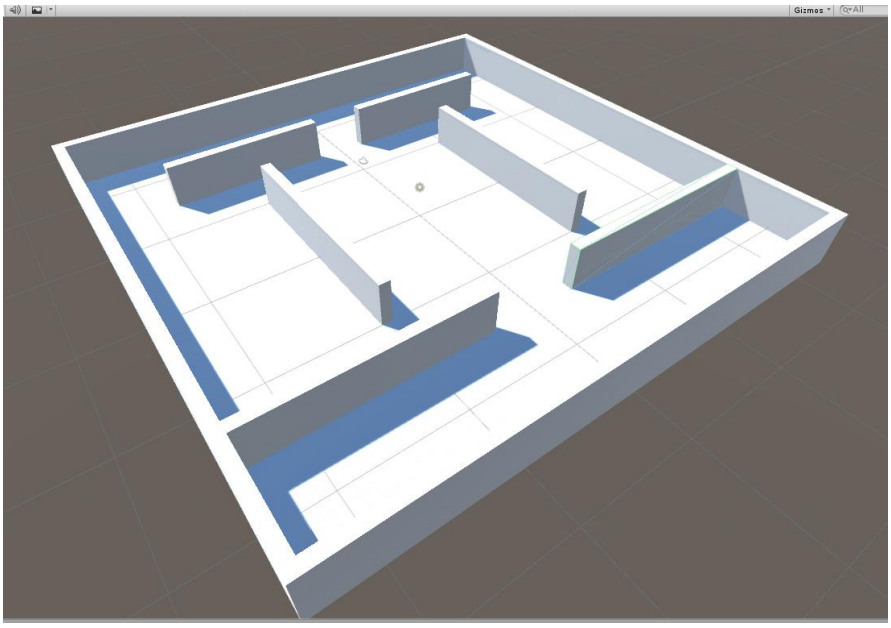

Figure 3: The Finished Scene with Outer and Inner Walls in Place

6. In the Hierarchical Tab, you probably noticed all the walls listed there. Wouldn't it be nice to show that they are linked together as a room (recall that the BoxPile from Lab 1 was made up of several Cubes)? To accomplish this, create an empty object. Go up to *GameObject -> Create Empty*. Rename this object to Room. **Make sure that the X, Y, and Z Positions are all 0!!!** Now, simply drag and drop each wall (and the floor) onto this object in the Hierarchical Tab to build the linkage.

Figure 4: Object Linkage to the Room

Tasks Part 2 – Let there be Light!!

1. When you created your empty scene, a directional light was already added. Select it in the Hierarchical Tab. After you have selected it, you'll notice the Inspector Tab will show its two components: the standard Transform Component and a Light Component. You want the directional light rather high above the scene, so *set the Y position to 15*. Nothing else needs to change in the Transform Component. You can rotate the light and see its effect. You don't want the light flat on the scene; the walls should produce some shadows to give a 3D effect. Thus, leaving the rotation on default is just fine. You will fill in the shadows with point lights.

2. There are a lot of settings in the Light Component most of which we will leave for now. The only setting you should change here is to reduce the Intensity (the brightness) down to 0.6. This will darken the scene so that the point lights will have more of an effect!

3. Now start placing the point lights. Follow the below steps:

    a. Go up to the menu and select *GameObject -> Light -> Point Light*.
    b. With the light object selected, set its X, Y, and Z position to 0 by editing the Position in the Transform component of the Inspector Tab.
    c. Change the Range in the Light Component of the Inspector Tab to 18. All other values can stay the same. You can ignore the warning about realtime indirect bounce lighting.
    d. Now move the light using the move button (refer to lab 1 if you forget how this button works). You want to put the light in areas of the scene that are dark, i.e. corners where the wall cast shadows.
    e. Repeat this until you have 3 to 5 point lights. Finally, place one more point light high above the scene (a Y position of 15) with a range of 40. This will add some variety to the light of the scene.
    f. Once finished, your scene should resemble something like Figure 5 (on the next page).

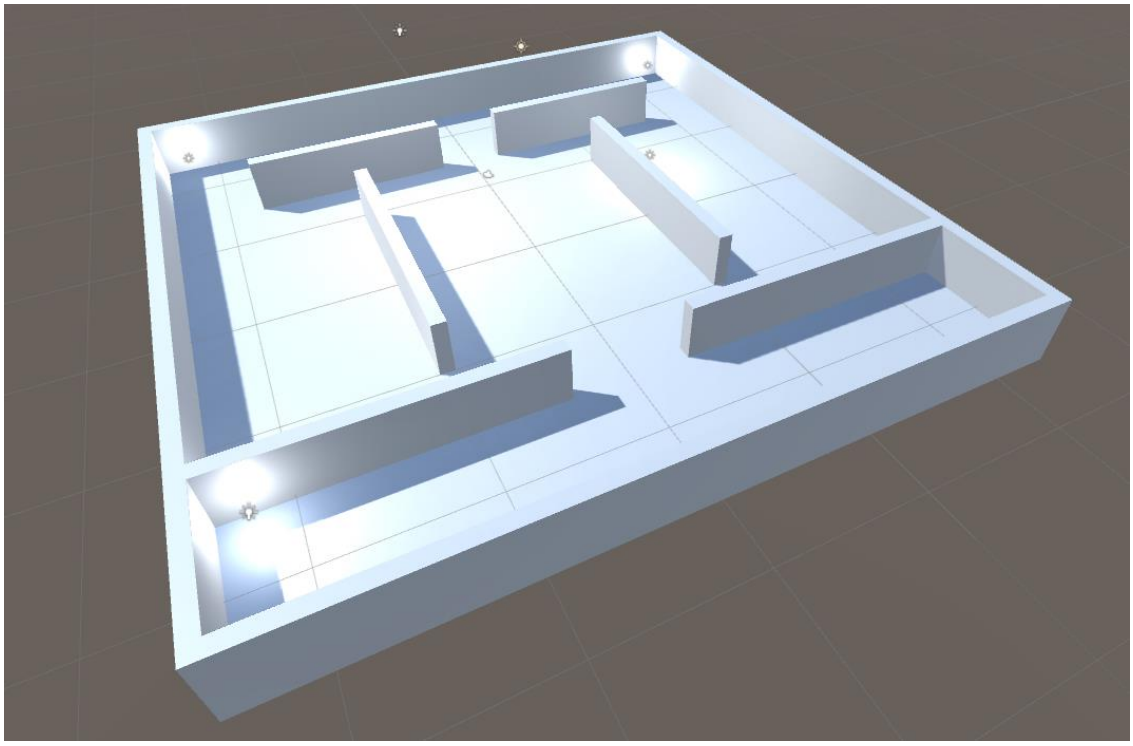

Figure 5: The Scene with Five Point Lights

Submission:

Demo Your Fully Lighted Scene (like that of Figure 5) to the Lab Instructor:

- **approx. 6 inner walls**
- **outerwalls should fully enclose floor**
- **no gaps between wall-to-wall or wall-to-floor connections**
- **Room linkage in Hierarchy Tab**
- **directional light at Y = 15**
- **3 to 5 point lights around the scene Incl. one at Y = 15**

- A successful demo scores you 10 marks.