

CS4109 – Computer Graphics

PBL 2

Submitted by:

Syed Hamid Haider	F20605034
Usman Malik	F20605053
Hassan Asif	F20605033

BSCS-F20 (7th Semester)

Submitted to:

Dr. Zulfiqar Ali

Dated:

Jan 7th, 2024



**Department of Computer Science
National University of Technology**

PBL Description

In this group PBL you are required to implement a basic ray tracer that can render a scene with accurate lighting, shadows, reflections, and refractions using Unity's graphics capabilities. You are provided with a basic scene setup in Unity consisting of the following elements and required tasks as following:

- a) A virtual camera positioned at a specific location and orientation within the scene.
- b) Several 3D objects such as spheres, cubes, and planes placed at different positions and orientations within the scene. Each object has its own material properties, including color, reflectivity, and transparency.
- c) One or more point or directional light sources emitting light rays that illuminate the scene. The light sources can have different intensities and colors.

Required Tasks

1. Set up a new Unity project and create a scene with the provided camera, objects, and light sources.
2. You are required to implement the ray casting algorithm to trace rays from the camera through each pixel of the screen and into the scene.
3. Implement intersection tests between the rays and the objects in the scene to determine which objects are visible from the camera's viewpoint.
4. Calculate the shading of each visible object based on its material properties (e.g., diffuse, specular, and ambient reflection) and the lighting conditions in the scene.
5. Implement shadow rays to determine if objects are in shadow, based on their visibility to the light sources.
6. Add support for reflections by recursively tracing reflected rays from reflective surfaces and combining the reflected color with the surface color. Render the final image of the scene by combining the colors calculated for each pixel.

Bonus Task

If you want to challenge yourself further, you can implement support for transparent materials and refraction effects using Unity's shader capabilities.

Implementation Strategy

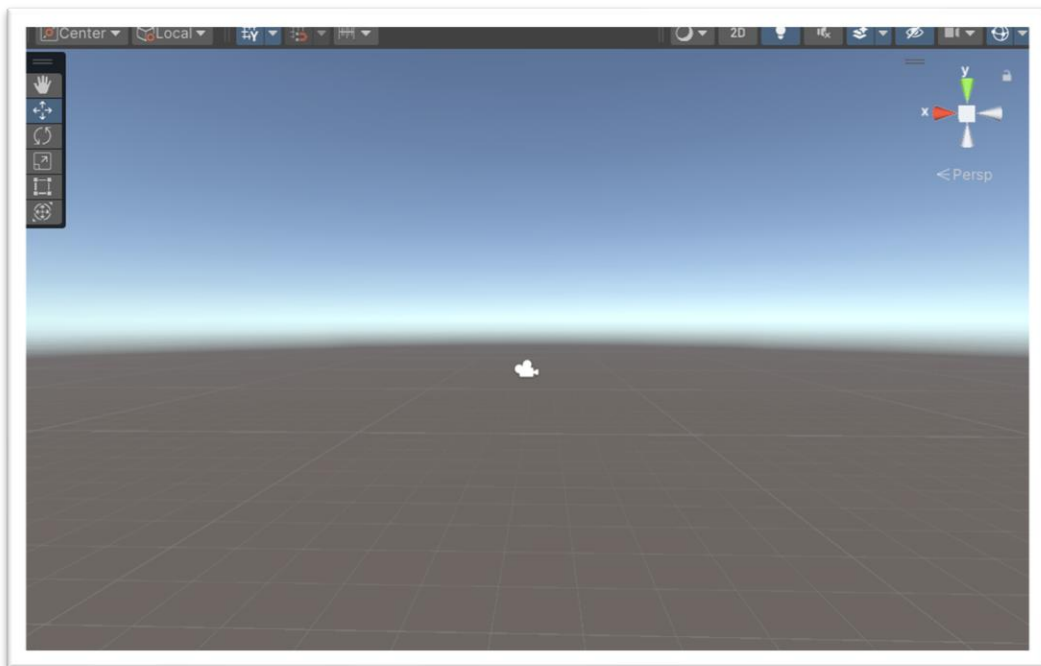
We have made the project in *Unity*. The *Ray Tracing* implementation, object creation, and a whole 3D scene is created using *Unity*'s 3D development engine.

We will follow the following steps while creating our 3D scene:

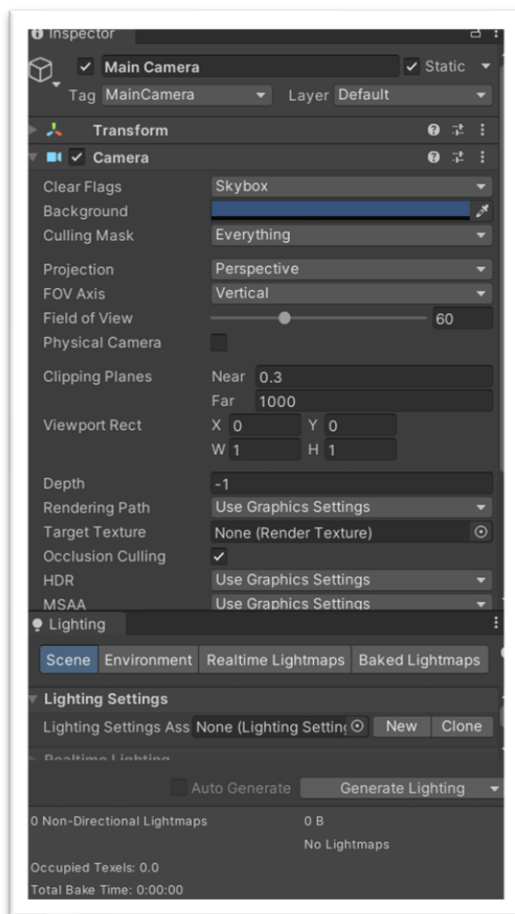
1. First, we set up the camera positioning on the plane.
2. Next, we set up the plane's color and properties. After setting up the plane, we set up the walls of the room that we have to set up. Each wall and (roof) is positioned according to the architecture to make a completed room. We set up the color, material and other properties of the walls while doing this as well.
3. When the room has been set up, next we have to add a light inside it so that the room can be lit up.
4. The next step is to add objects in the room to populate the room and create a complete 3D scene.
5. The first object we add is a sphere of yellow color. We also set a *Reflection Probe* for this sphere so that we can see the reflection in the scene. This is where *Ray Tracing* comes into play.
6. Similarly, we set up more objects like blue capsule, red sphere, and green cube, and we set the *Reflection Probes* for each.
7. After we have finished setting up all the objects, we can try to see the effects of light shining on them by turning the light set up inside the room earlier on or off. We will clearly be able to see the difference in the effect of reflection on the objects when the light is turned on or off. This is *Ray Tracing* into play.
8. Finally, we have finished setting up our 3D scene, we can try to run the *Unity* project to see how it might look like in a game.

Next, we will show the screenshots of the *Unity* scene development process.

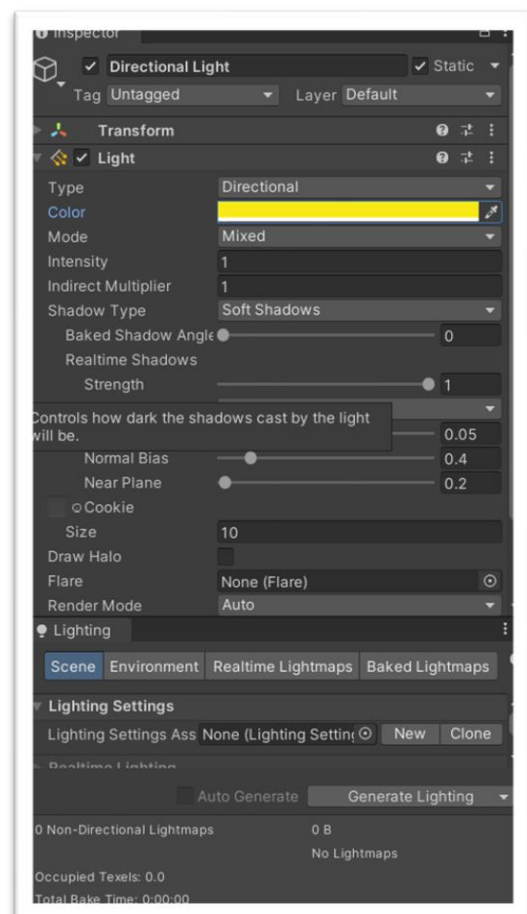
Implementation Screenshots



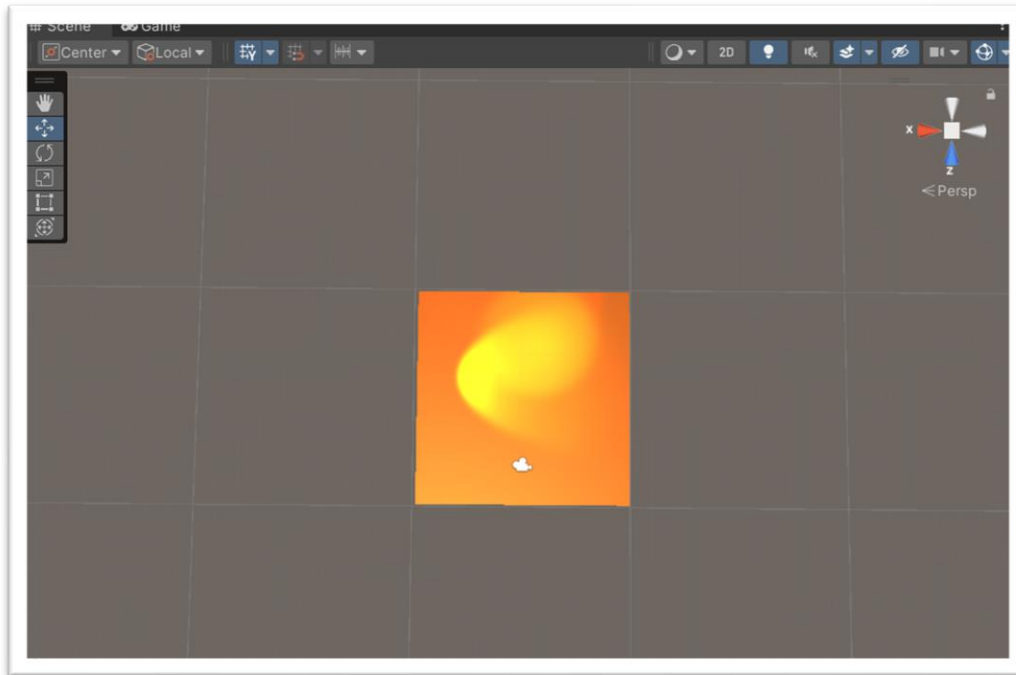
Positioning the Camera On the 3D Space



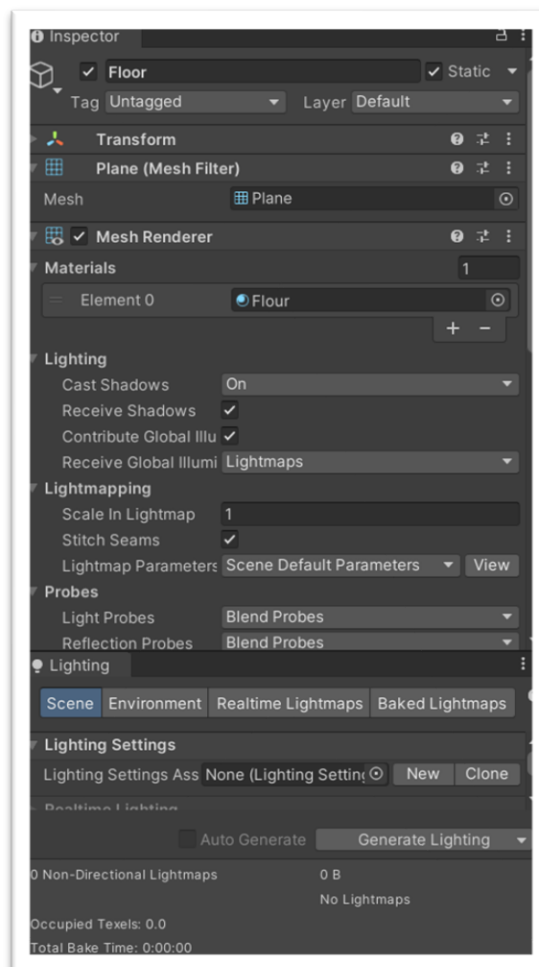
Main Camera Settings



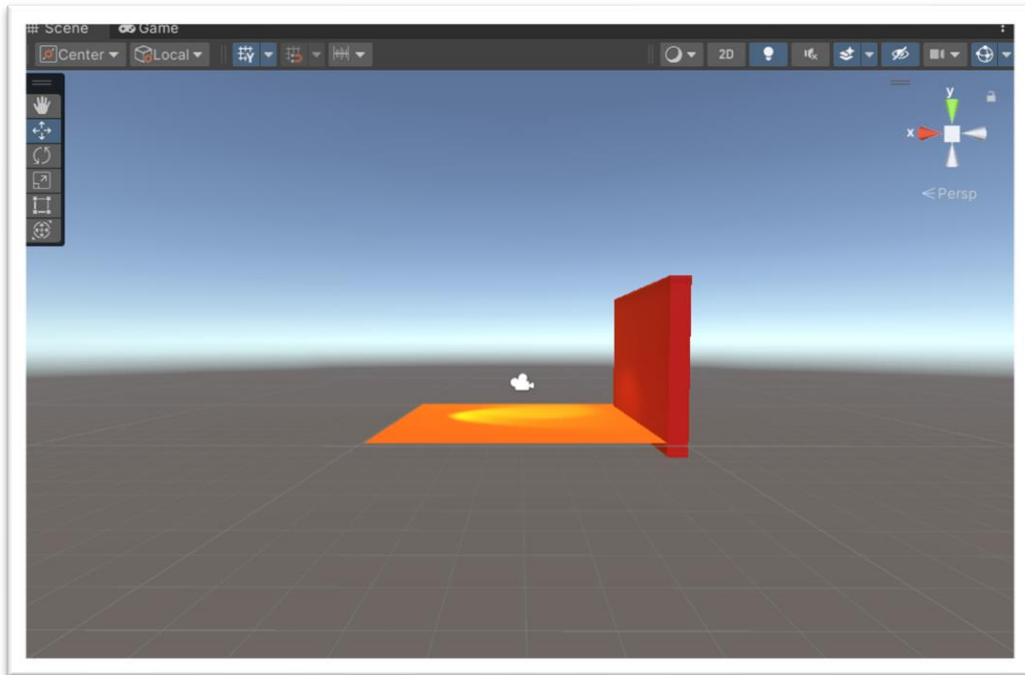
Main Camera Directional Light Settings



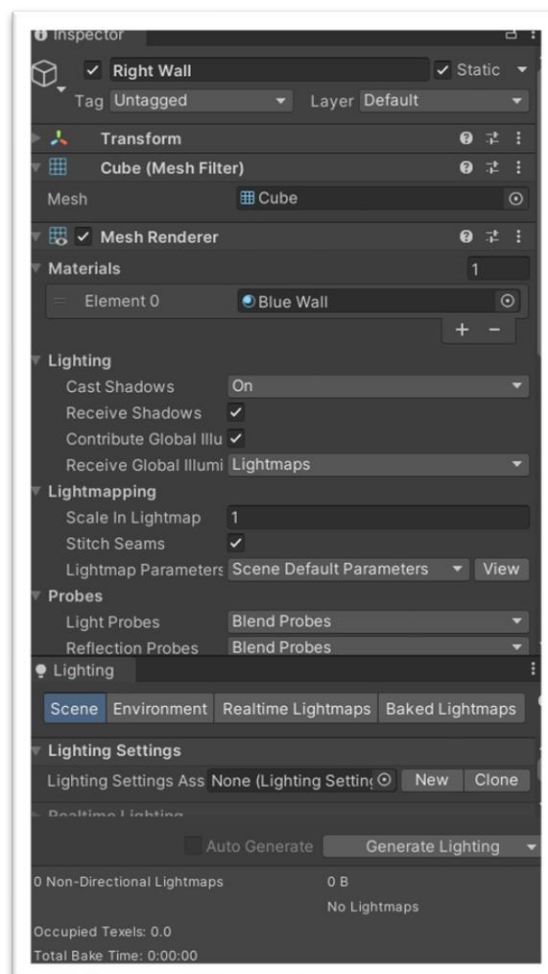
Setting Up the Plane in the 3D Space (Top View)



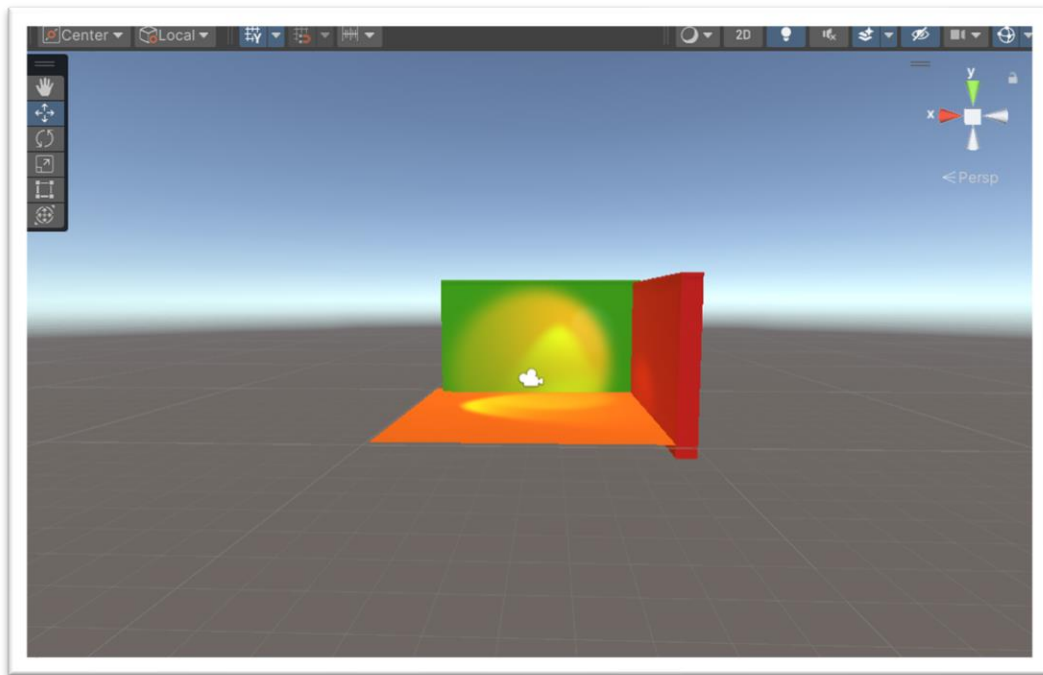
Properties for the Plane Set Up



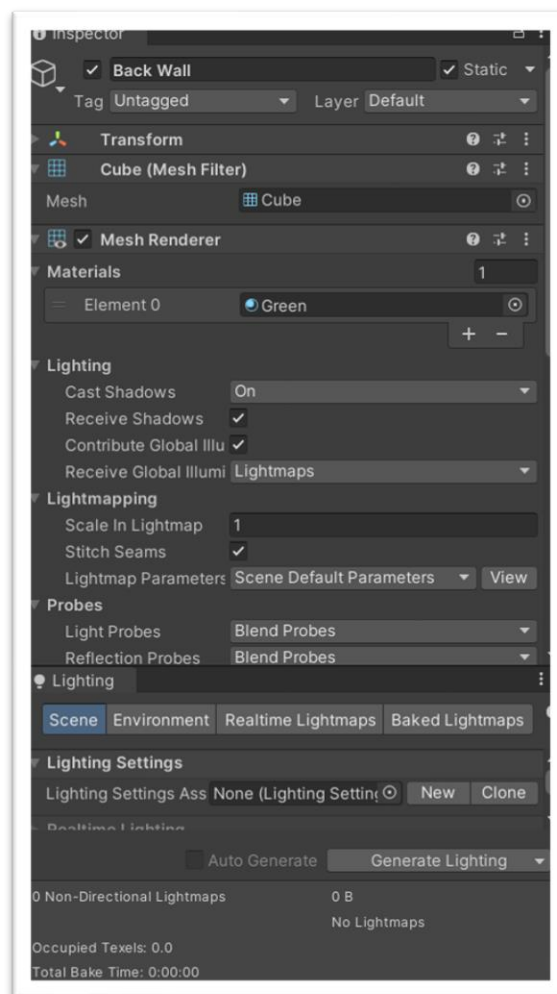
Setting up the Right Wall



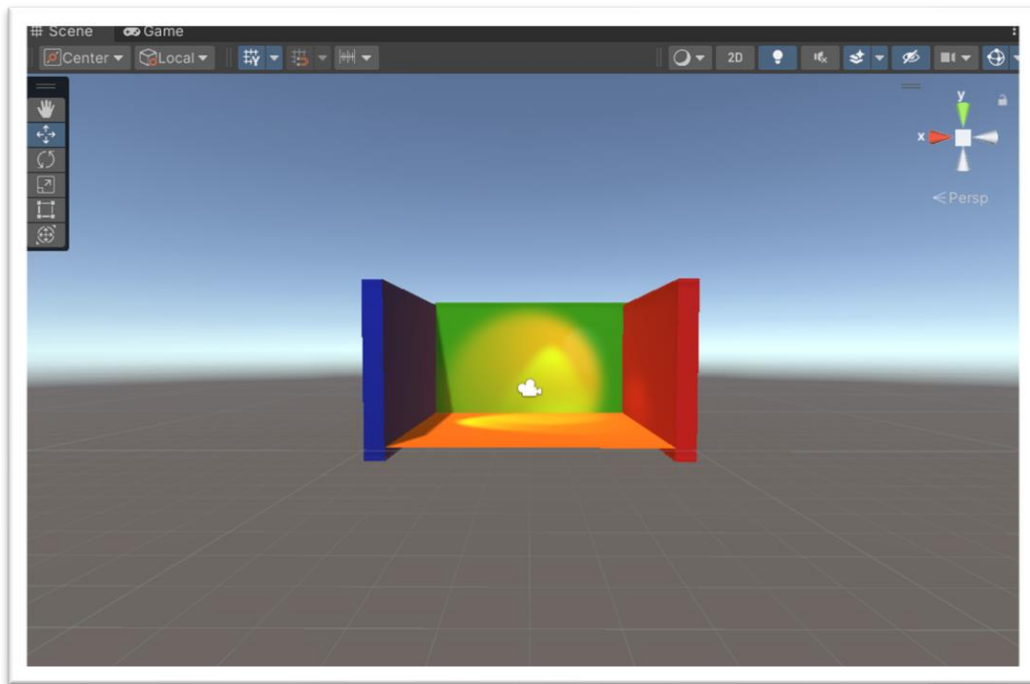
Right Wall Properties



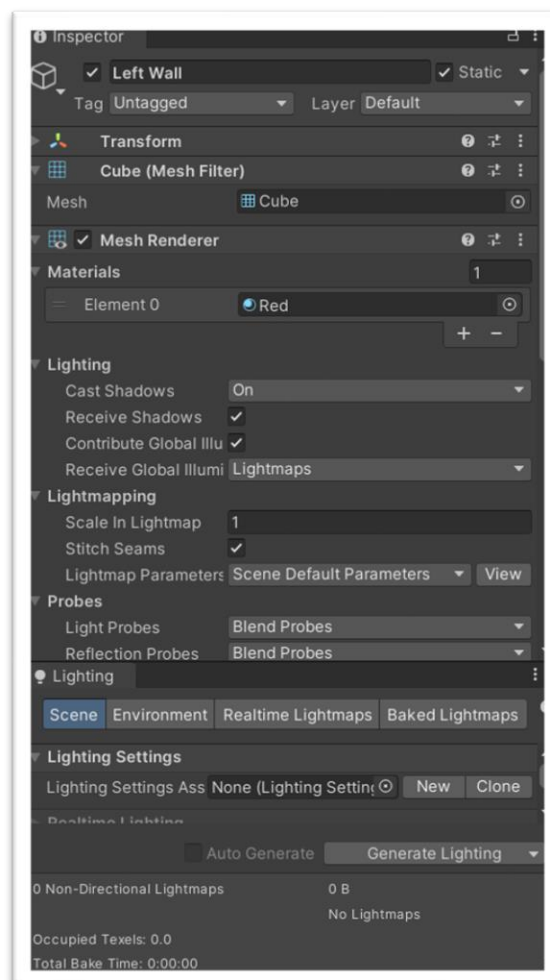
Setting Up the Back Wall



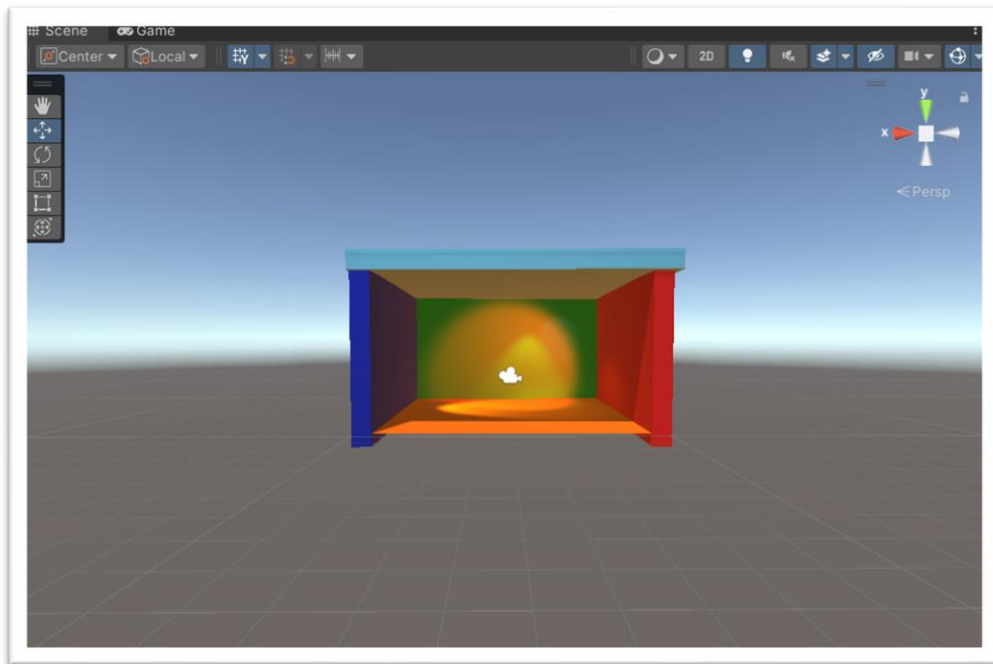
Back Wall Properties



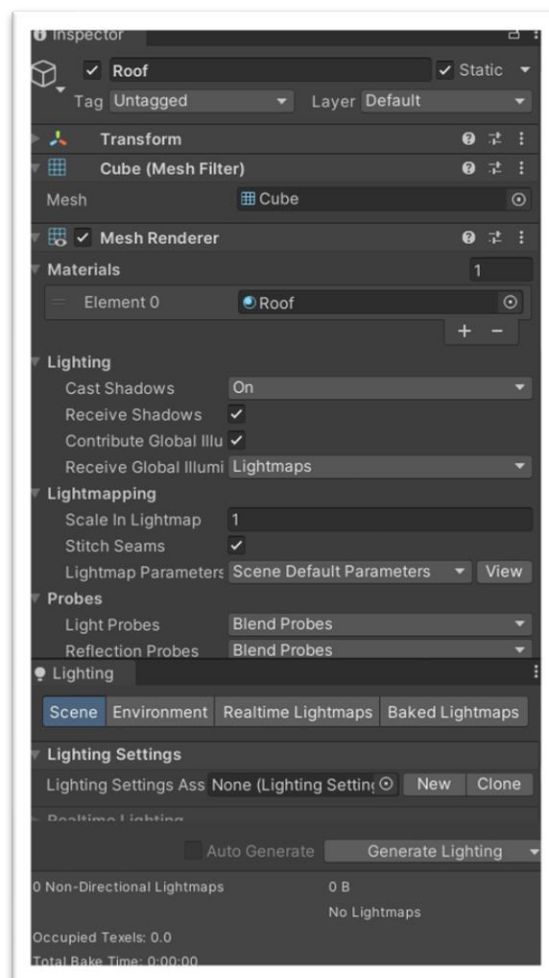
Setting Up the Left Wall



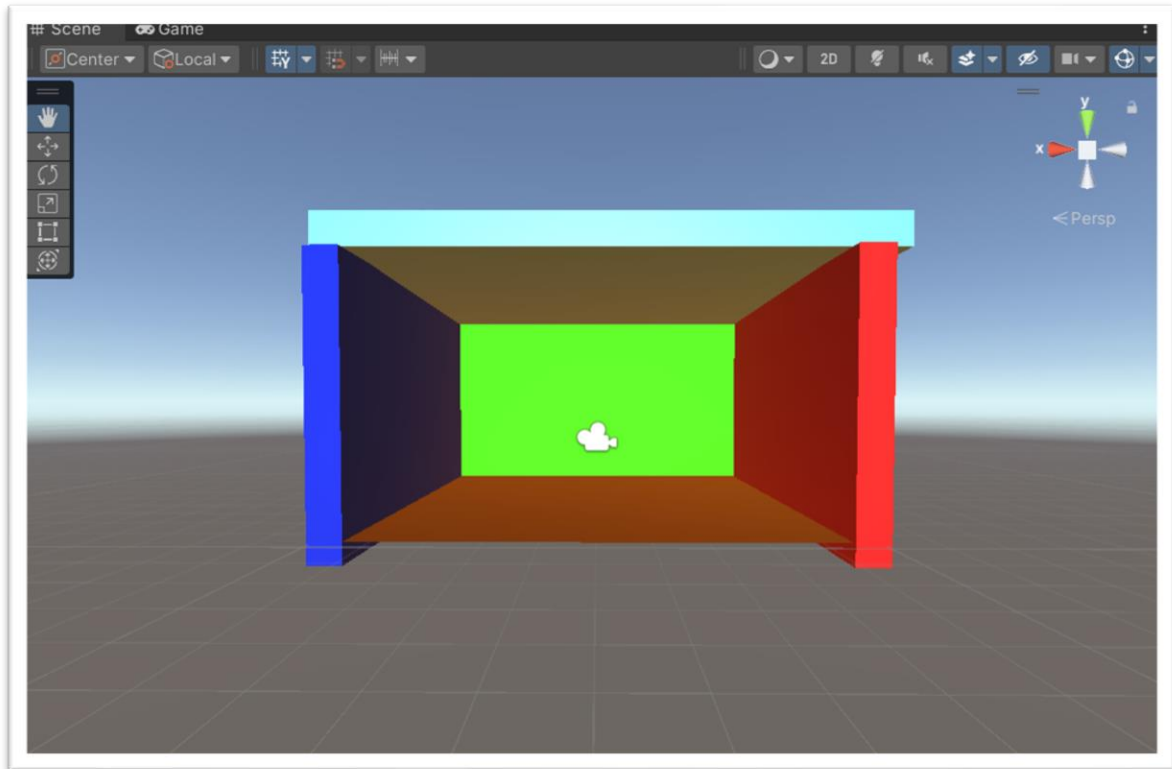
Left Wall Properties



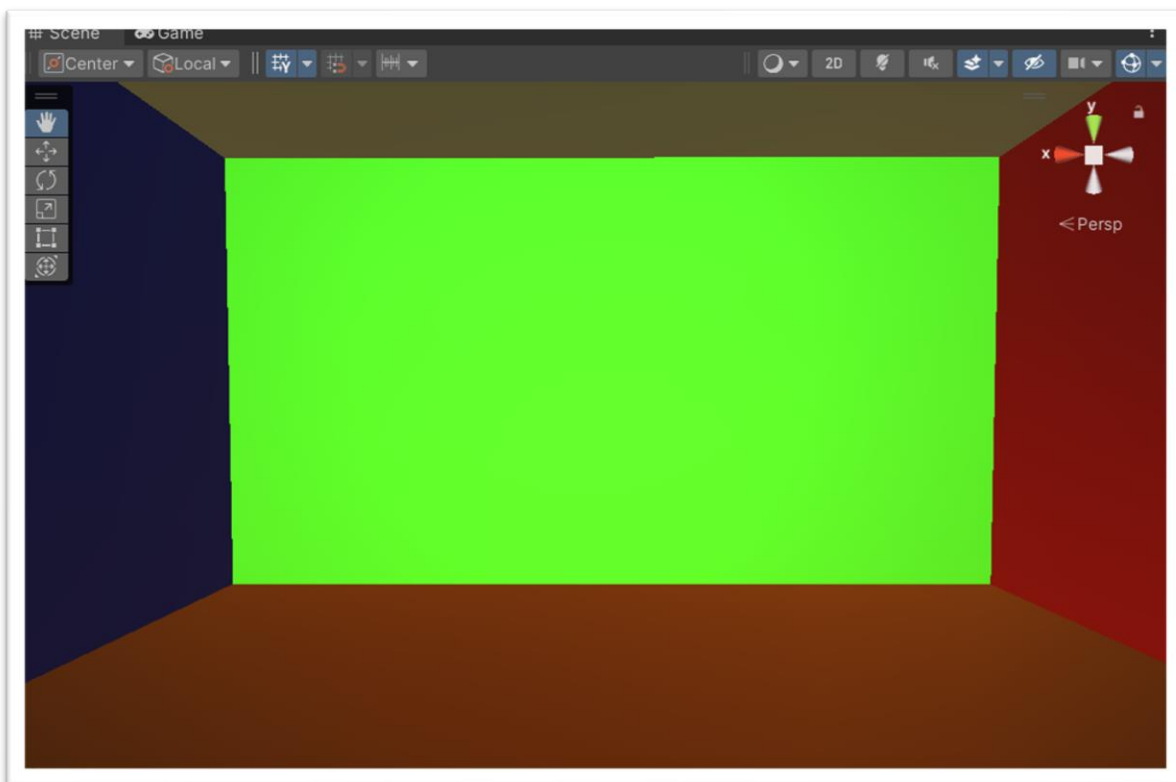
Setting Up the Roof



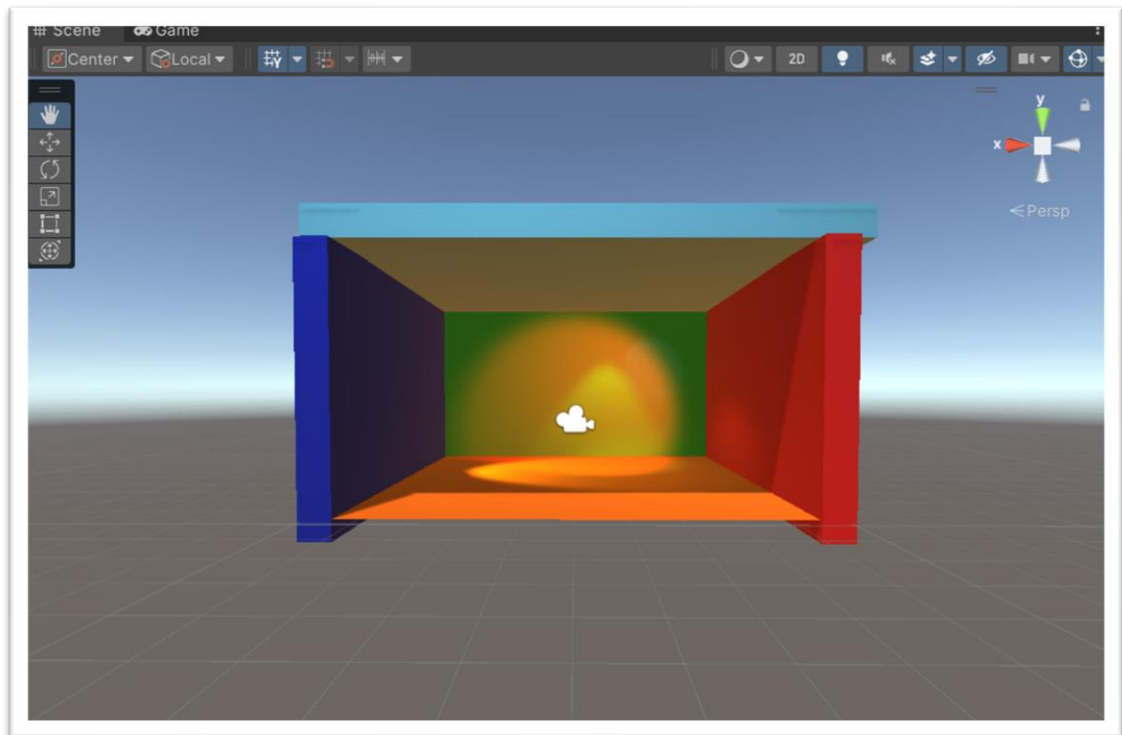
Roof Properties



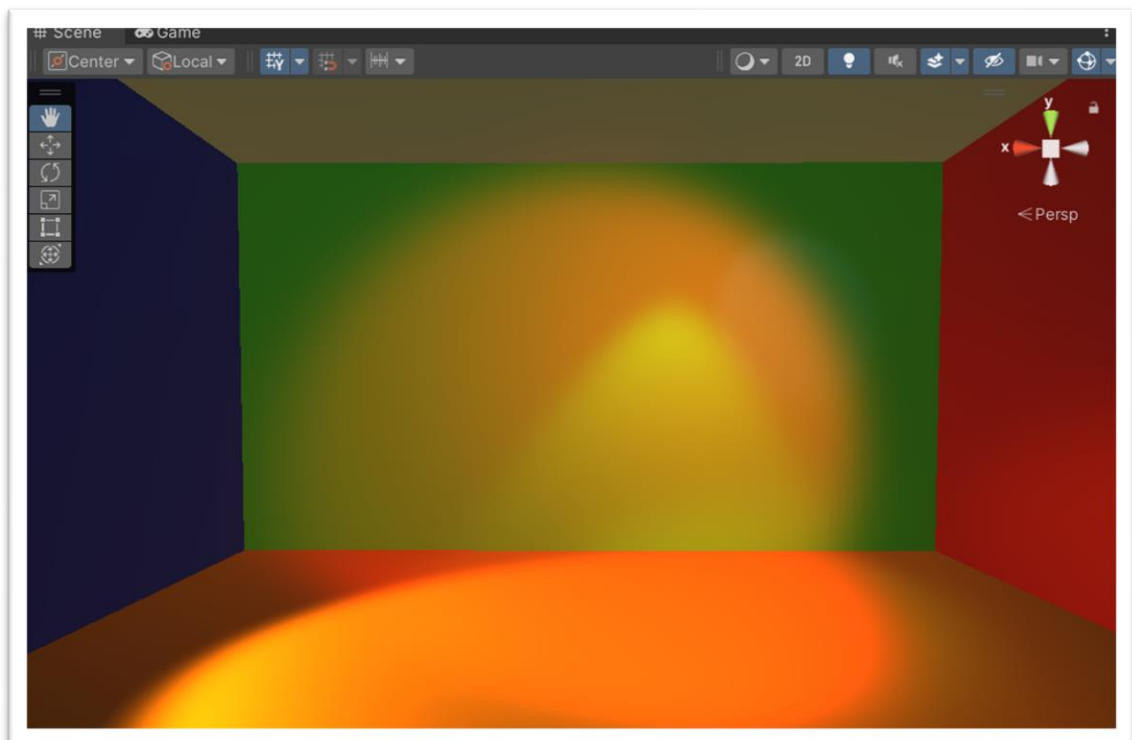
Room Zoomed Out When Directional Light is Off



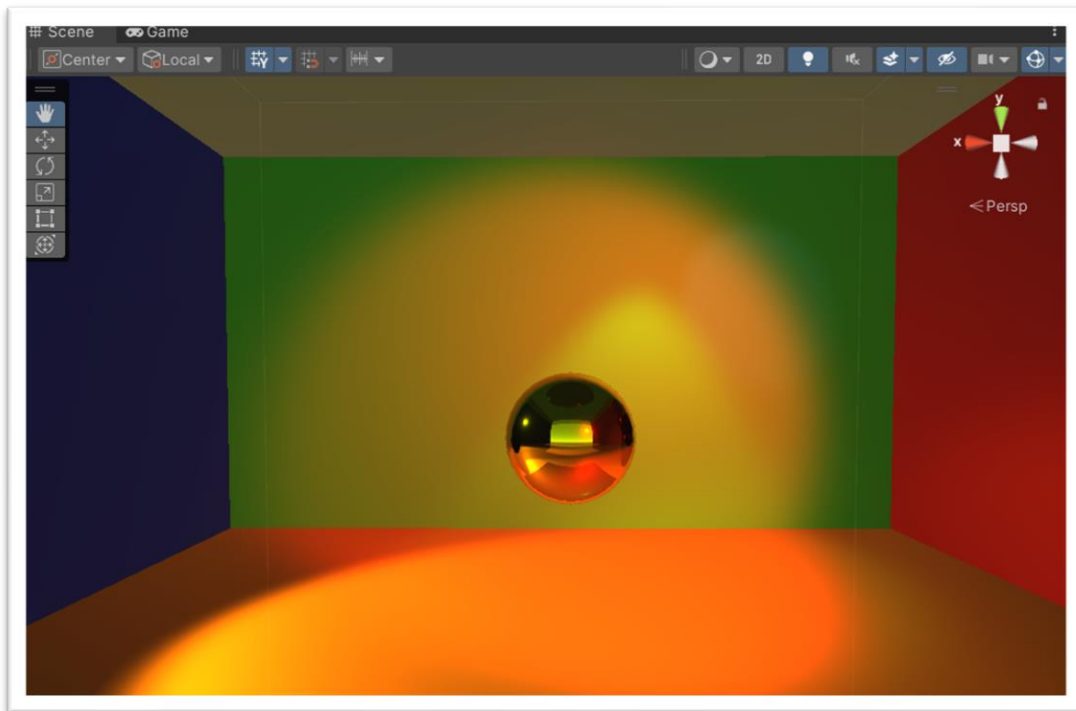
Room Zoomed In When Directional Light is Off



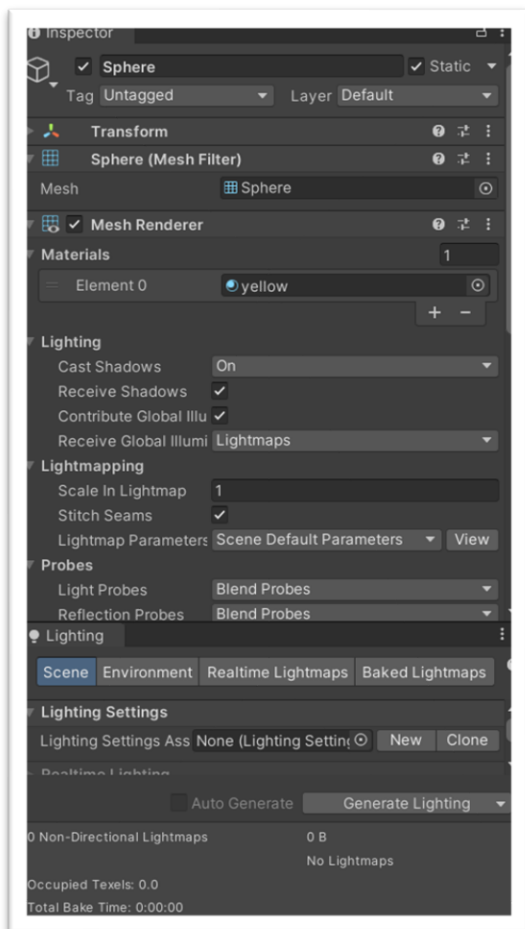
Room Zoomed Out When Directional Light is On



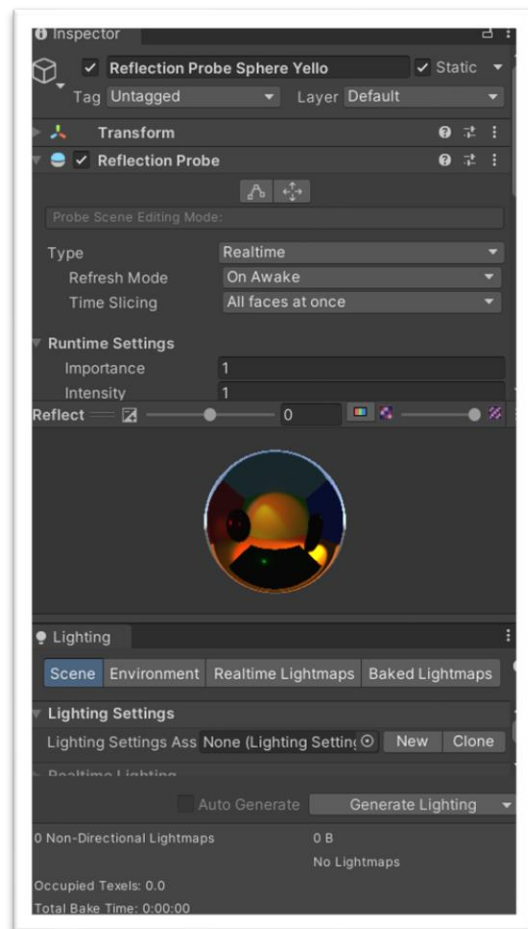
Room Zoomed In When Directional Light is On



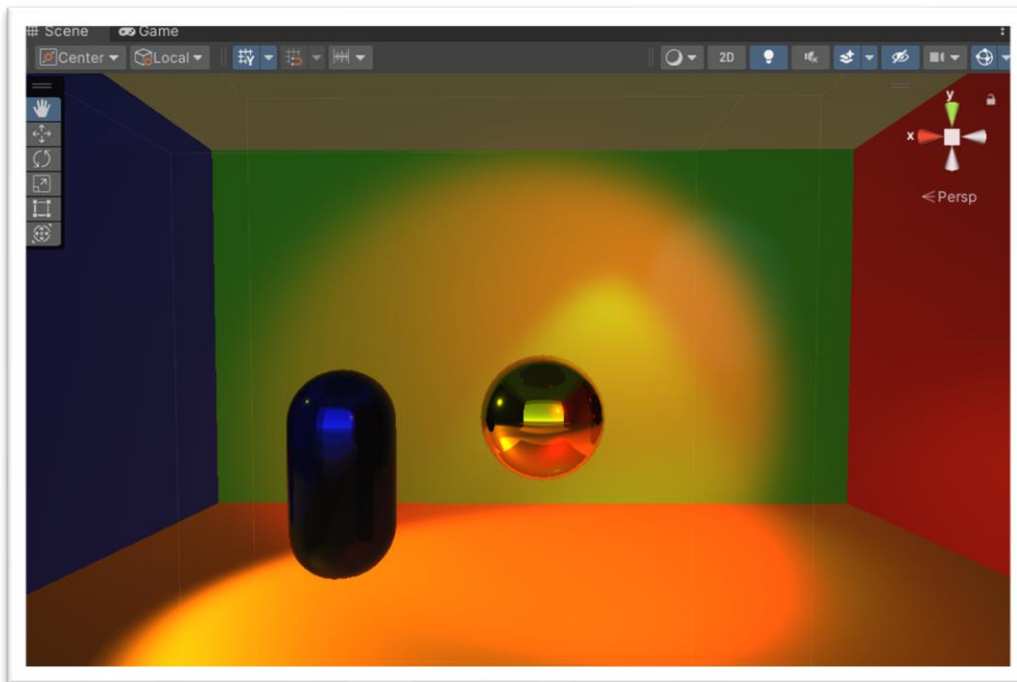
Setting Up a Yellow Sphere in the Room



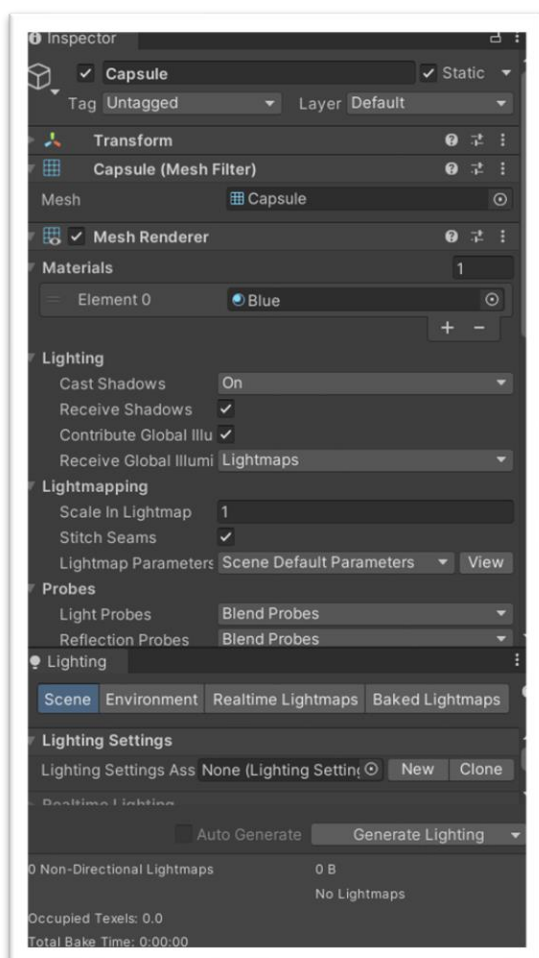
Yellow Sphere Properties



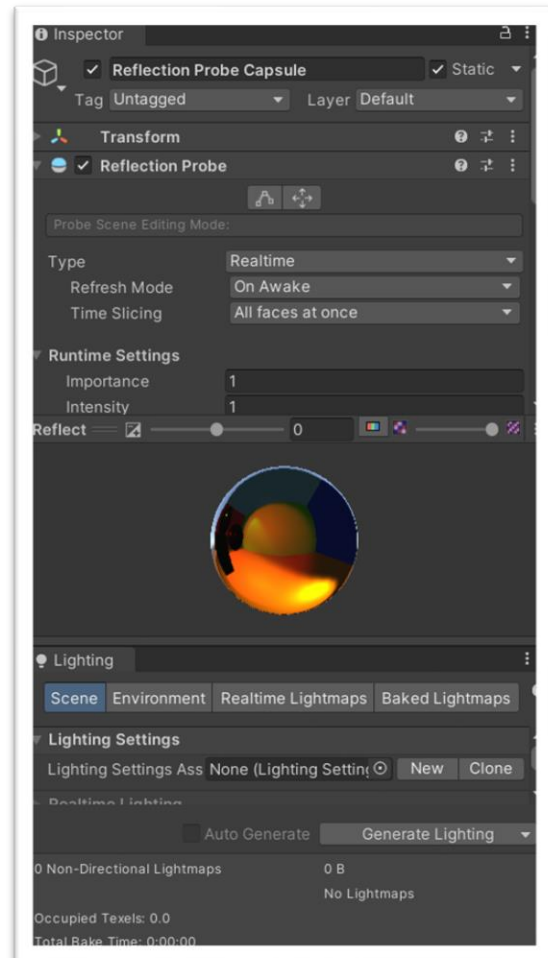
Sphere Reflection Probe Settings



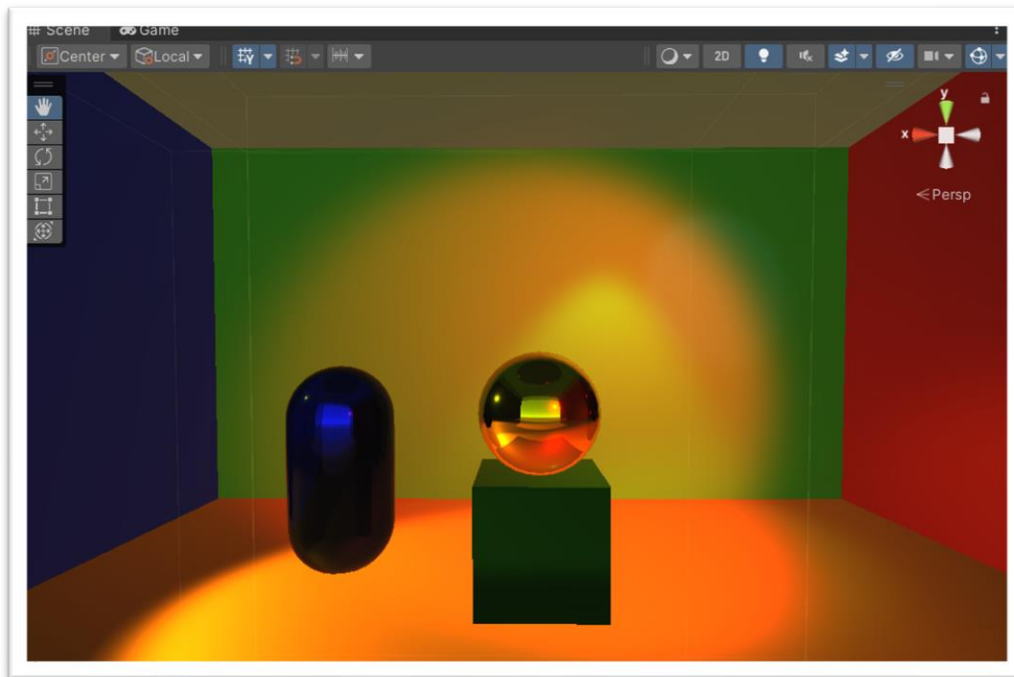
Adding a Blue Capsule Shape in the Room



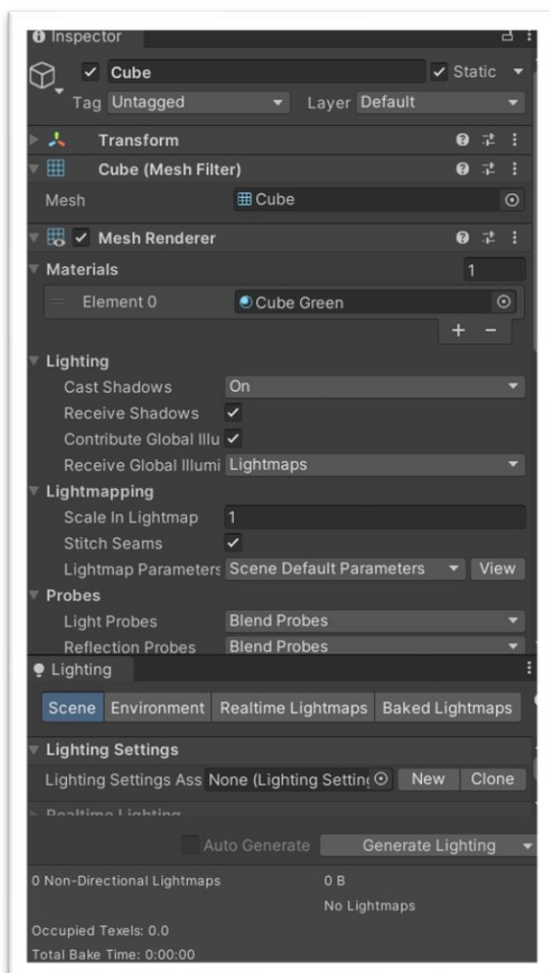
Blue Capsule Properties



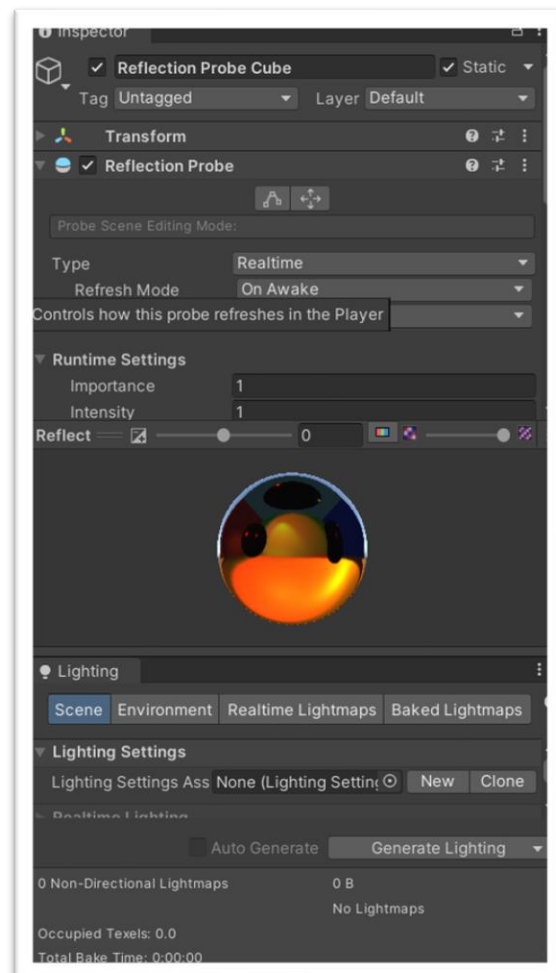
Blue Capsule Reflection Probe Settings



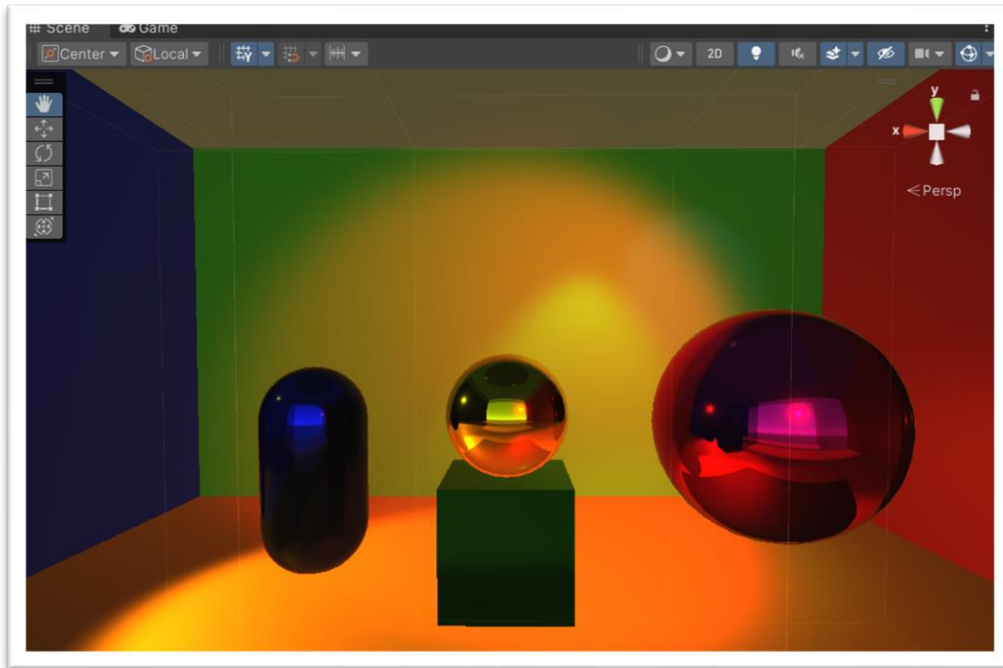
Adding a Green Cube in the Room



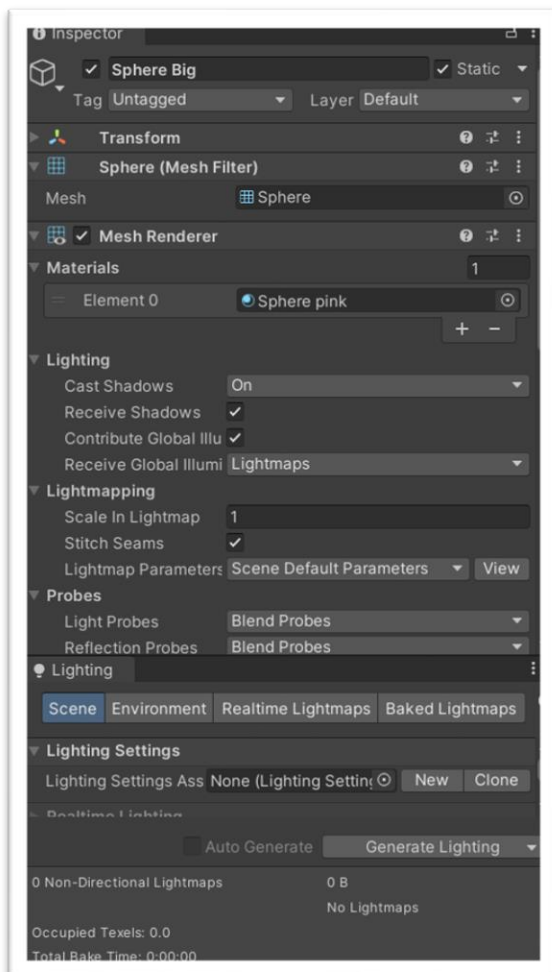
Green Cube Properties



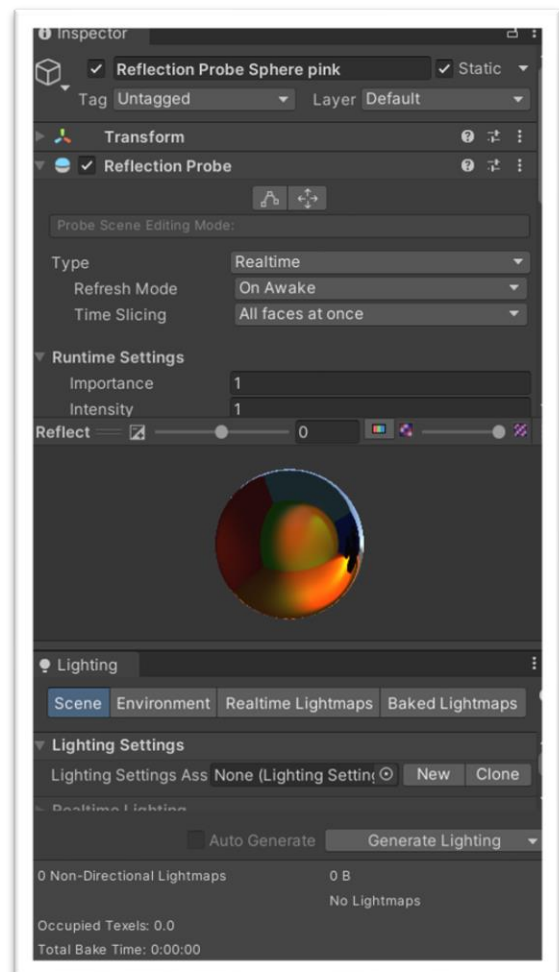
Green Cube Reflection Probe Settings



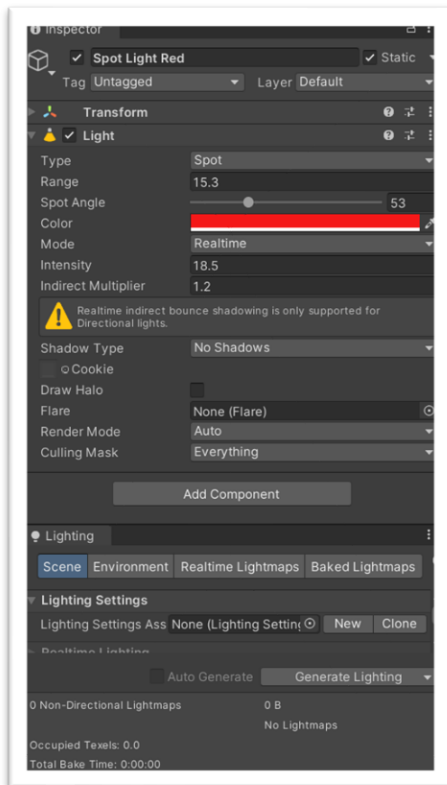
Adding Another Sphere (Red) in the Room



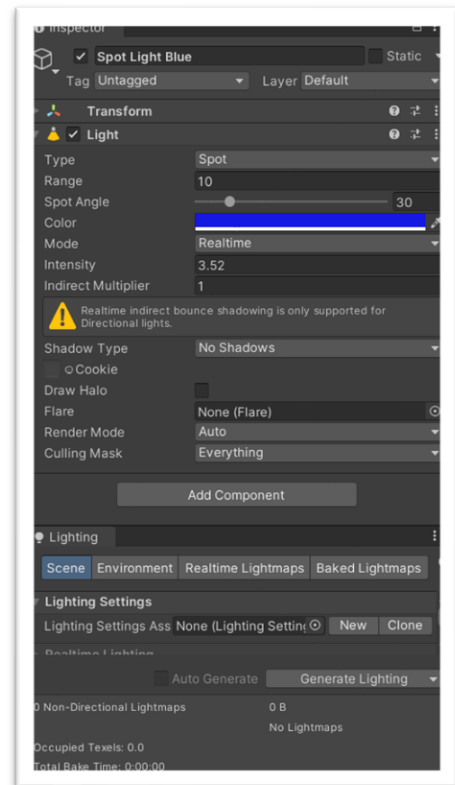
Red Sphere Properties



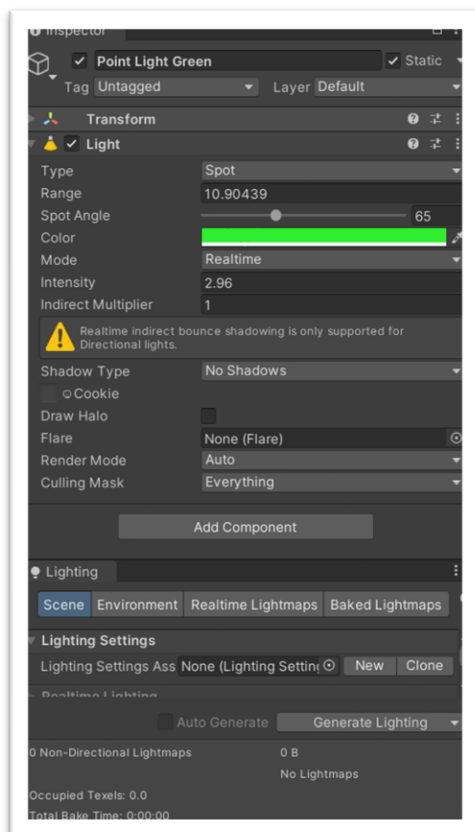
Red Sphere Reflection Probe Settings



Red Spotlight Properties



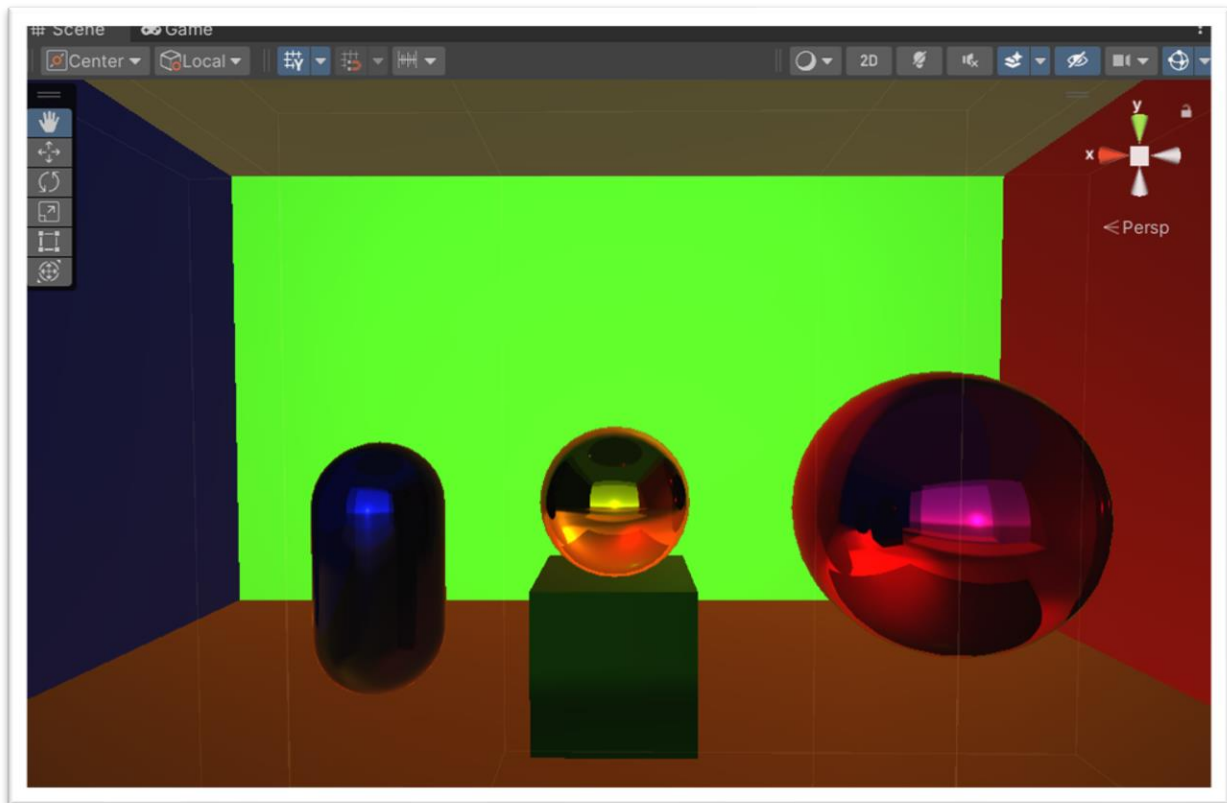
Blue Spotlight Properties



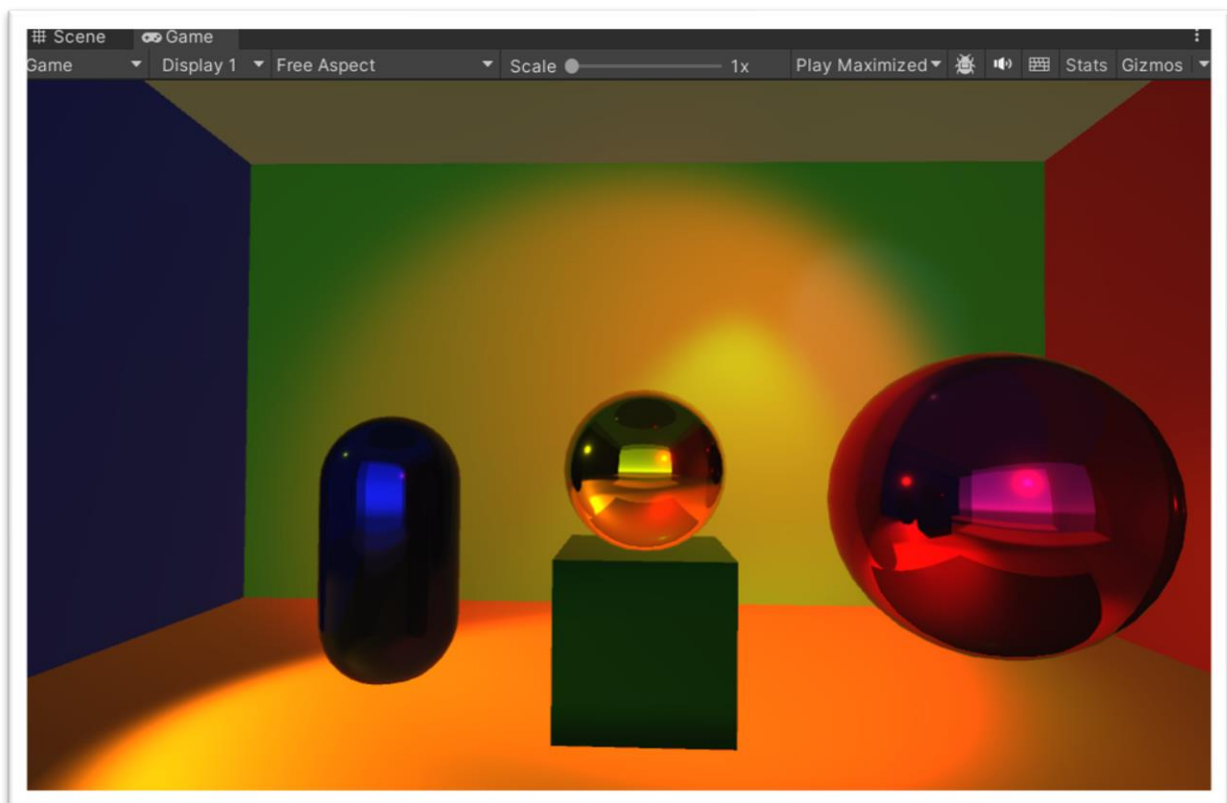
Green Point Light Properties



Color Given to All the Materials



Final 3D Scene When the Directional Light is Turned Off



Final Scene When the Directional Light is Turned On

Conclusion

- In this PBL, we have made a 3D scene in *Unity*.
- The scene has four walls, a roof, a yellow sphere, a blue capsule, a red sphere, and a green cube. All of the objects (not walls or roof) have been configured with *Ray Tracing* algorithms to give off reflections.
- There is also the directional lighting from the camera that lights up objects. Furthermore, there is red and blue spotlighting, and also a point light.
- The combined effect of all the objects, room, and lighting gives the result of a realistic 3D scene with objects placed around the room.
- We can see in the screenshots the final result of our project.

What We Have Learned

- We have learned how to set up a scene in *Unity*.
 - We have learned how to use lighting in *Unity*, including spotlights, and point lights.
 - We have learned how to use *Ray Tracing* and make the objects reflective.
 - Lastly, we have learned how to add colors and shading on the objects in *Unity*.
-

[END]