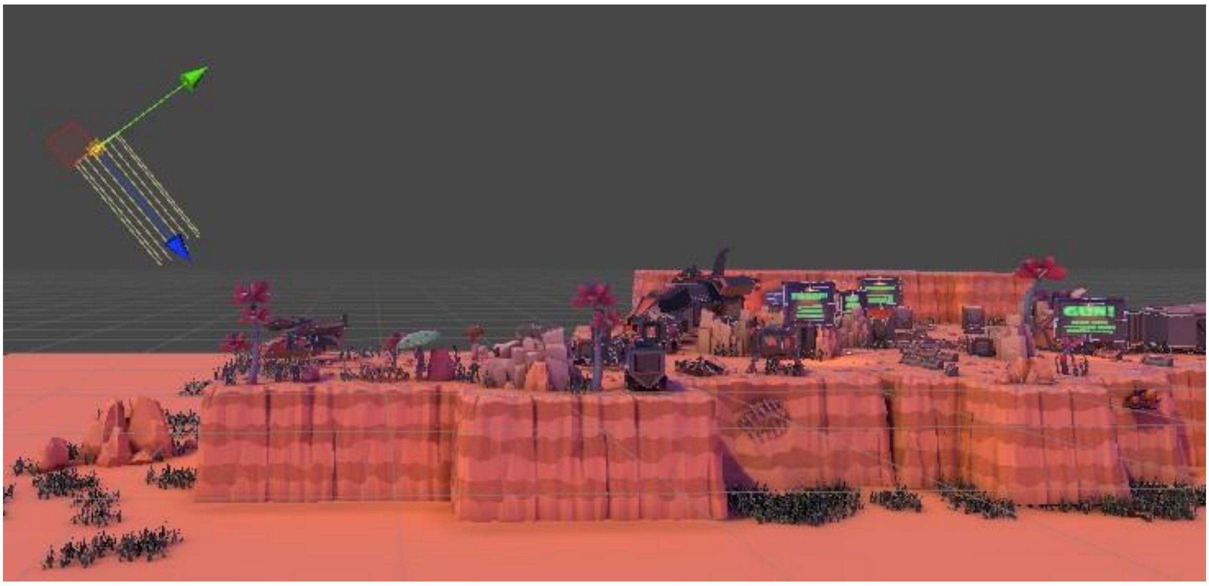
Sprint 03

Assignment 01: Lighting & VFX

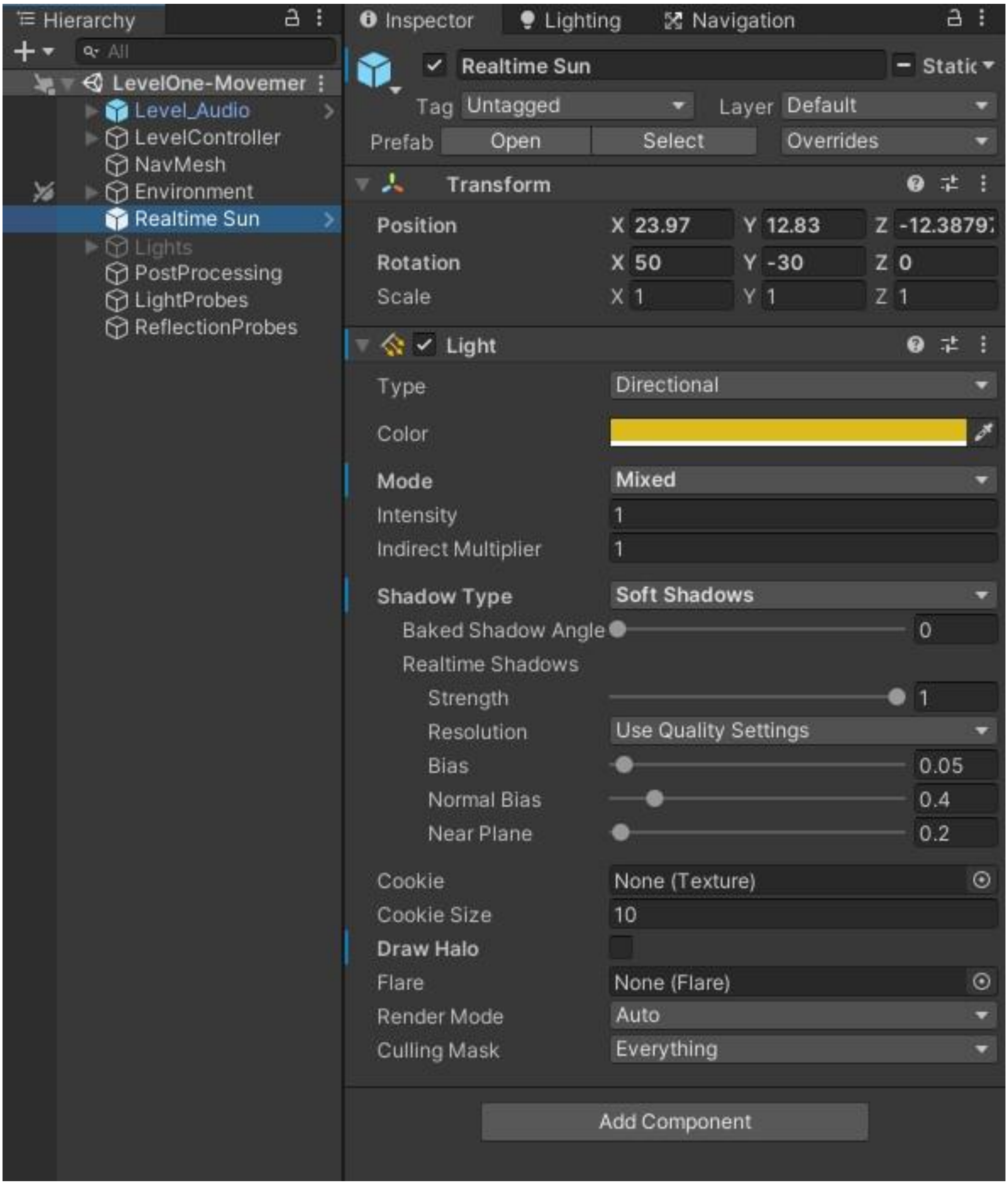
Step 01: Setup

- ❖ **Directional Light.**

- We open the “LevelOne-Movement” scene.
- Then we begin with the setup.
- We have the “Directional Light” by default.
- We just change its mode to “Mixed” & enable shadows as “Soft Shadows”.



- ❖ **Light Probes, Reflection Probes & Ambient Occlusion.**

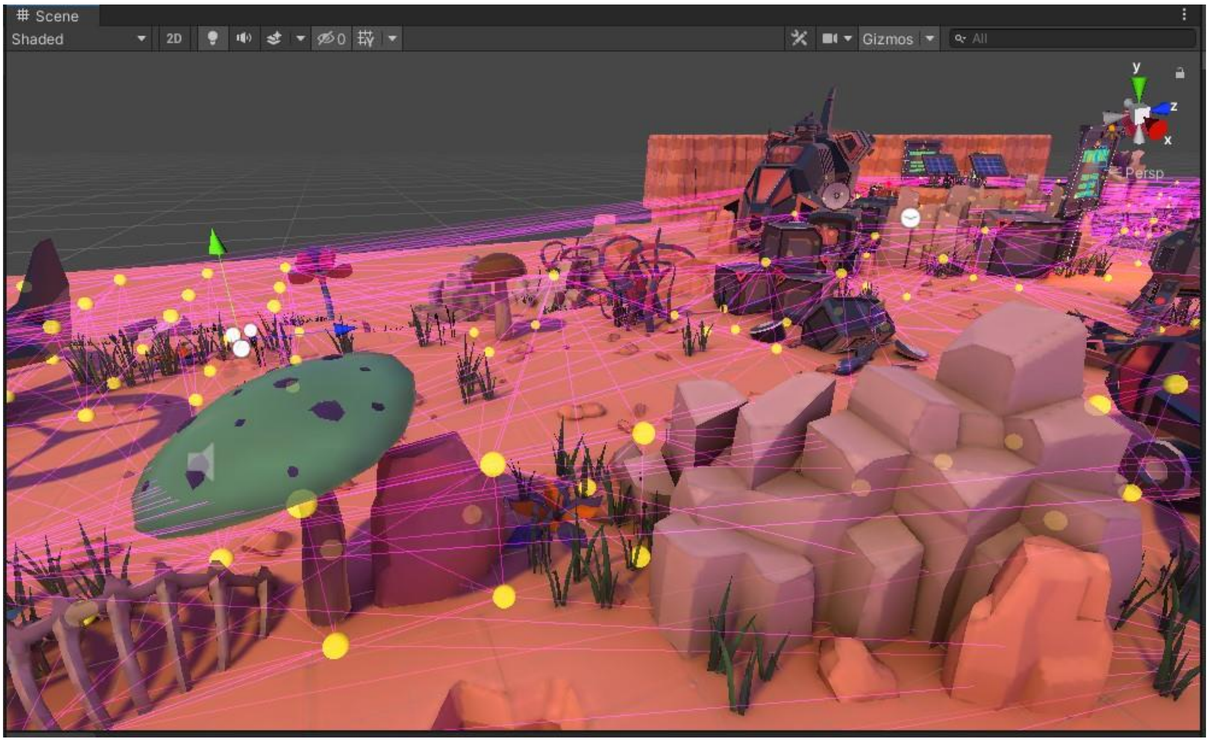


Realtime Sun
Tag Untagged
Layer Default

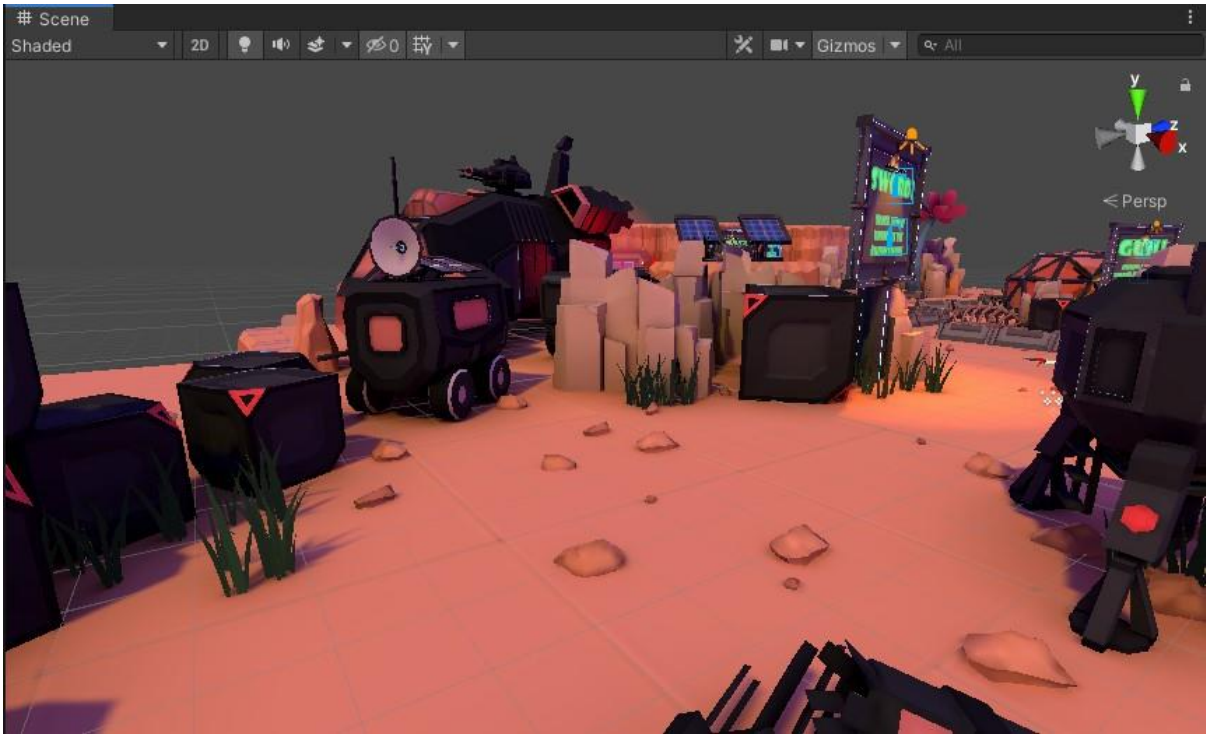
Transform
Position X 23.97 Y 12.83 Z -12.3879
Rotation X 50 Y -30 Z 0
Scale X 1 Y 1 Z 1

Light
Type Directional
Color
Mode Mixed
Intensity 1
Indirect Multiplier 1
Shadow Type Soft Shadows
Baked Shadow Angle 0
Realtime Shadows
Strength 1
Resolution Use Quality Settings
Bias 0.05
Normal Bias 0.4
Near Plane 0.2
Cookie None (Texture)
Cookie Size 10
Draw Halo ☐
Flare None (Flare)
Render Mode Auto
Culling Mask Everything

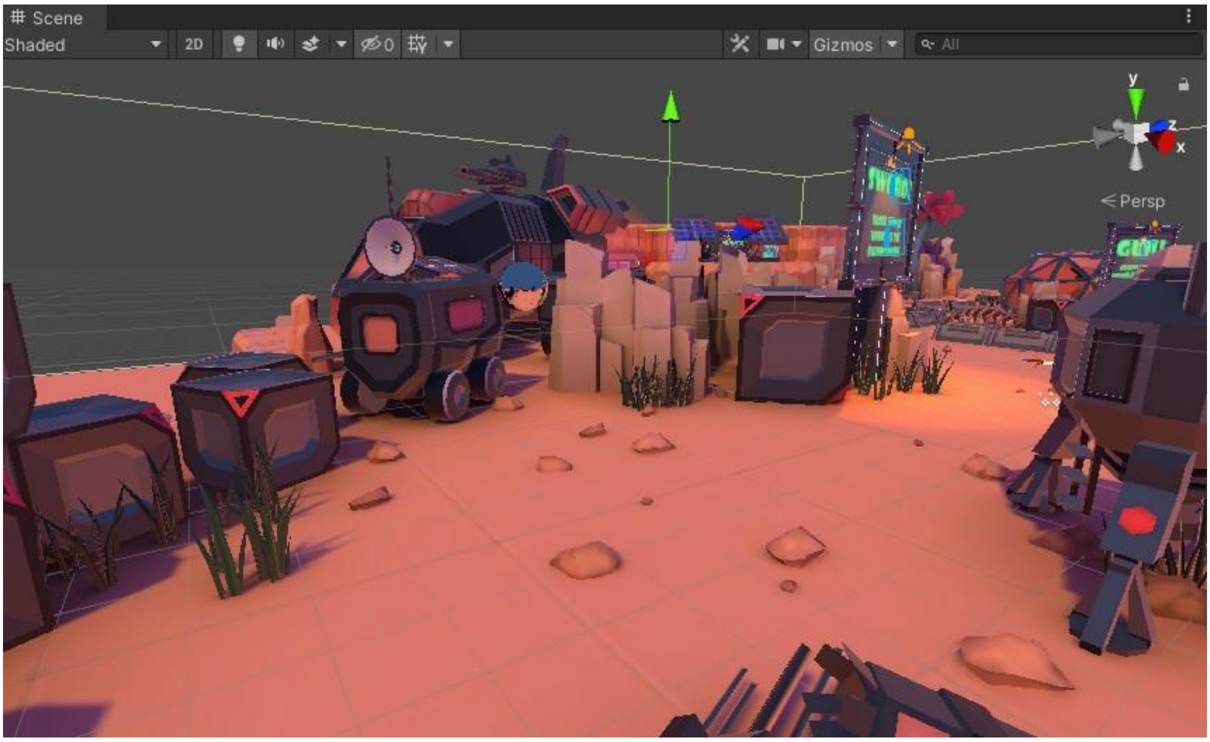
- Then we create an “Empty GameObject” and add “LightProbeGroup” component to it.
- Then we “Edit” the Light Probes and Duplicate them and place them in the scene wherever required.



- Similar way, we create "ReflectionProbe" by creating an Empty GameObject and adding the "ReflectionProbe" component to it.
- Then we adjust the Bounding-Box of the Reflection Probe and place the Sphere at an appropriate position.

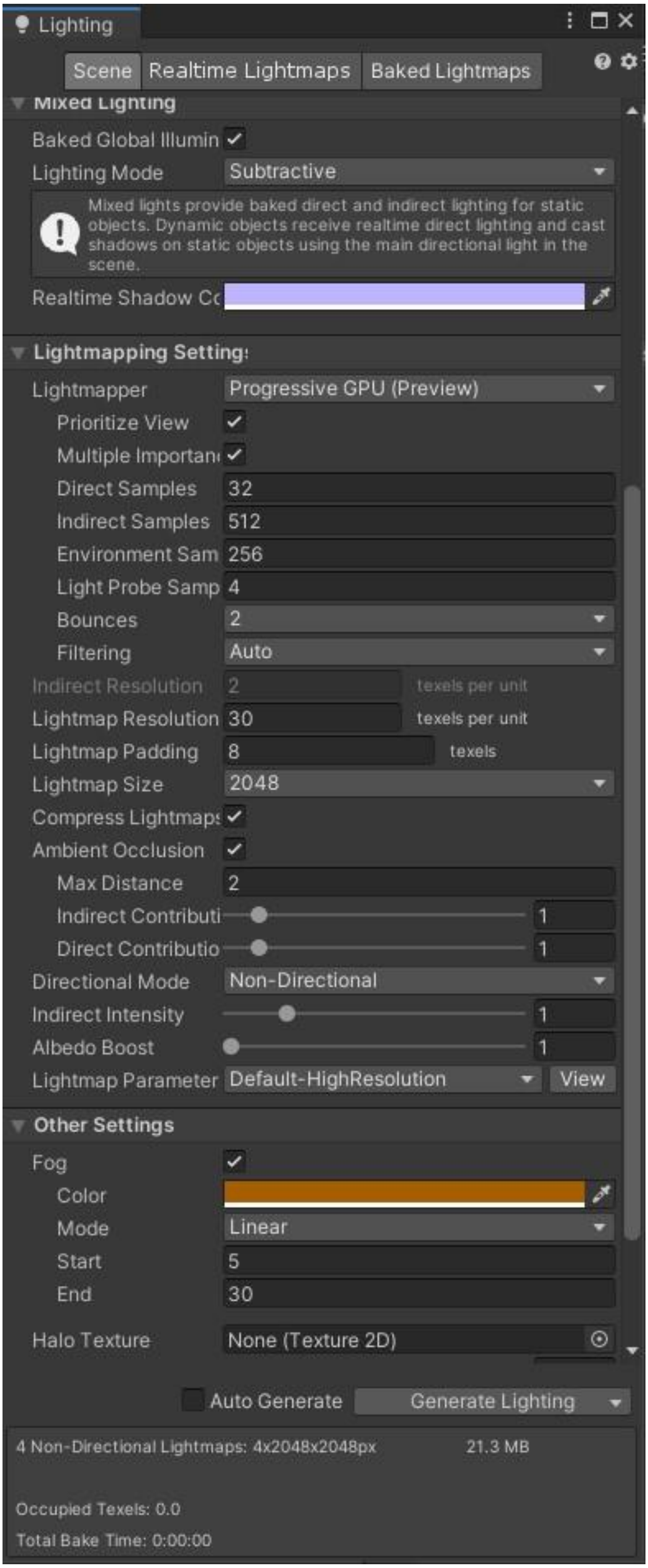


Before (No Reflection Probe)



After (Adding Reflection Probe)

- Then we go into the “Lighting Settings” (Windows > Rendering > Lighting Settings).
- Then we add the appropriate values in the properties and make sure that the “AmbientOcclusion” checkbox is ON.
- Then we bake the LightMap.
- We also click the “Fog” checkbox and add some Fog into the scene with appropriate settings.



Mixed Lighting

Baked Global Illumin ✓

Lighting Mode Subtractive

Mixed lights provide baked direct and indirect lighting for static objects. Dynamic objects receive realtime direct lighting and cast shadows on static objects using the main directional light in the scene.

Realtime Shadow Cc

Lightmapping Setting:

Lightmapper Progressive GPU (Preview)

Prioritize View ✓

Multiple Importan ✓

Direct Samples 32

Indirect Samples 512

Environment Sam 256

Light Probe Samp 4

Bounces 2

Filtering Auto

Indirect Resolution 2 texels per unit

Lightmap Resolutio 30 texels per unit

Lightmap Padding 8 texels

Lightmap Size 2048

Compress Lightmap: ✓

Ambient Occlusion ✓

Max Distance 2

Indirect Contributi 1

Direct Contributio 1

Directional Mode Non-Directional

Indirect Intensity 1

Albedo Boost 1

Lightmap Parameter Default-HighResolution View

Other Settings

Fog ✓

Color

Mode Linear

Start 5

End 30

Halo Texture None (Texture 2D)

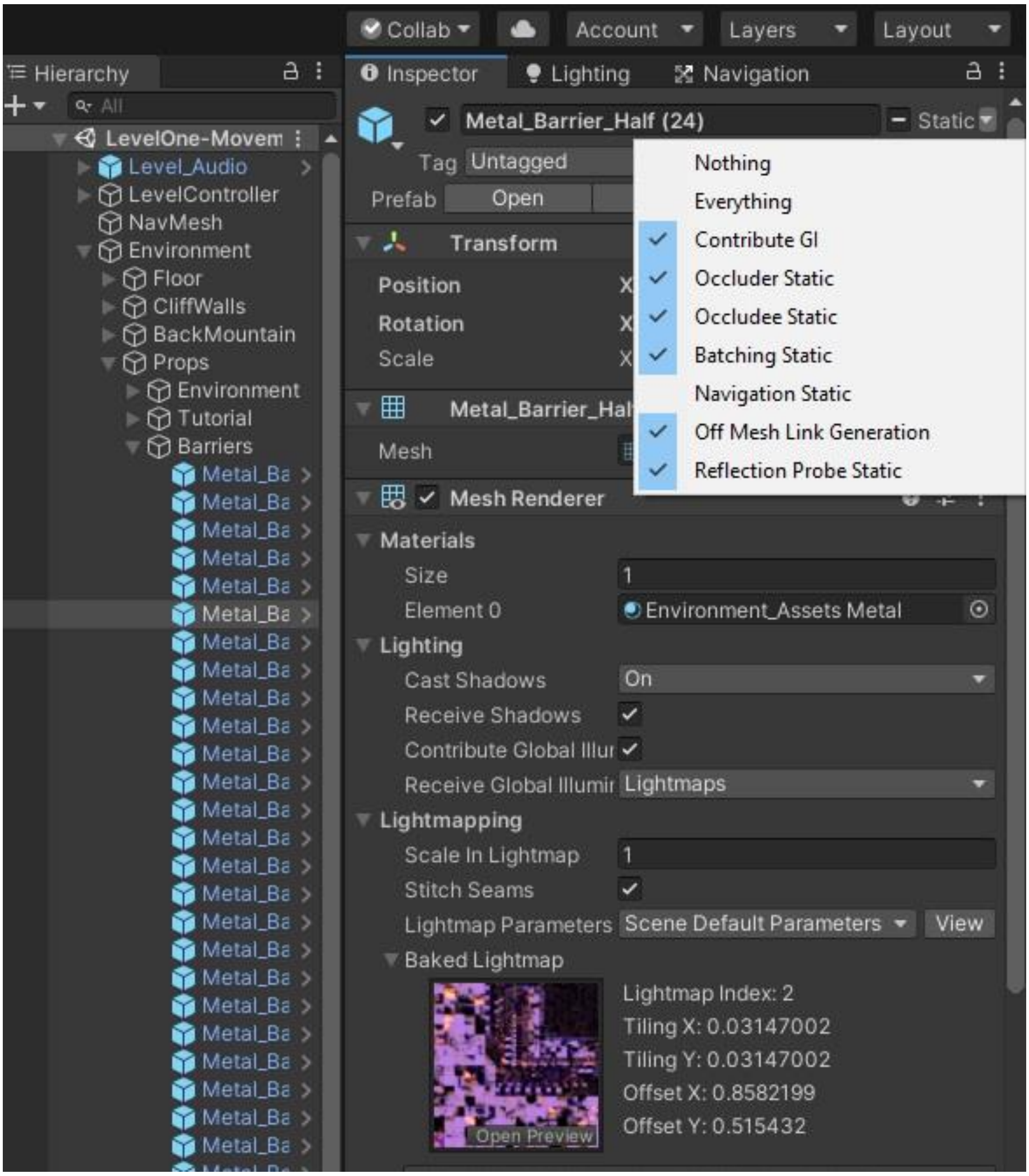
□ Auto Generate Generate Lighting

4 Non-Directional Lightmaps: 4x2048x2048px 21.3 MB

Occupied Texels: 0.0

Total Bake Time: 0:00:00

- Then set the “Static Flags” for each Object present in the scene to contribute to “Contribute GI and other attributes”.



☑ Metal_Barrier_Half (24) Static

Tag Untagged

Prefab Open

Nothing
Everything
✓ Contribute GI
✓ Occluder Static
✓ Occludee Static
✓ Batching Static
Navigation Static
✓ Off Mesh Link Generation
✓ Reflection Probe Static

Position X
Rotation X
Scale X

Metal_Barrier_Hal
Mesh

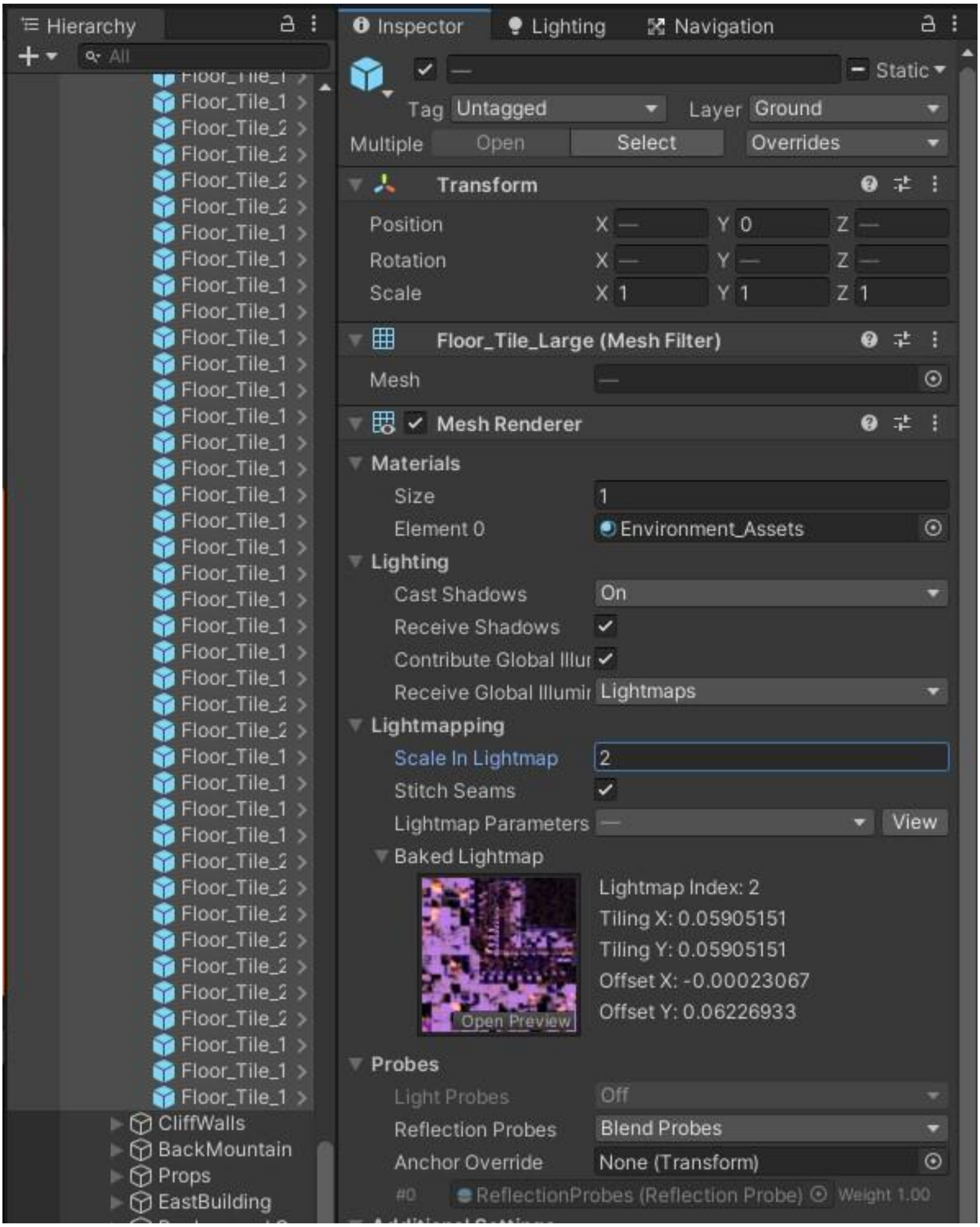
Materials
Size 1
Element 0 Environment_Assets Metal

Lighting
Cast Shadows On
Receive Shadows ☑
Contribute Global Illur ☑
Receive Global Illumir Lightmaps

Lightmapping
Scale In Lightmap 1
Stitch Seams ☑
Lightmap Parameters Scene Default Parameters

Baked Lightmap
Lightmap Index: 2
Tiling X: 0.03147002
Tiling Y: 0.03147002
Offset X: 0.8582199
Offset Y: 0.515432

- Then set the “LightMap Scale” values for every gameobject in the scene.
- We only change the LightMap Scale value of all the “Floor” gameobjects to “2” and rest we leave it with the default values.



Tag Untagged Layer Ground

Multiple

Transform

Position X — Y 0 Z —

Rotation X — Y — Z —

Scale X 1 Y 1 Z 1

Floor_Tile_Large (Mesh Filter)

Mesh —

Mesh Renderer ☑

Materials

Size 1

Element 0 Environment_Assets

Lighting

Cast Shadows On

Receive Shadows ☑

Contribute Global Illur ☑

Receive Global Illumir Lightmaps

Lightmapping

Scale In Lightmap 2

Stitch Seams ☑

Lightmap Parameters —

Baked Lightmap

Lightmap Index: 2
Tiling X: 0.05905151
Tiling Y: 0.05905151
Offset X: -0.00023067
Offset Y: 0.06226933

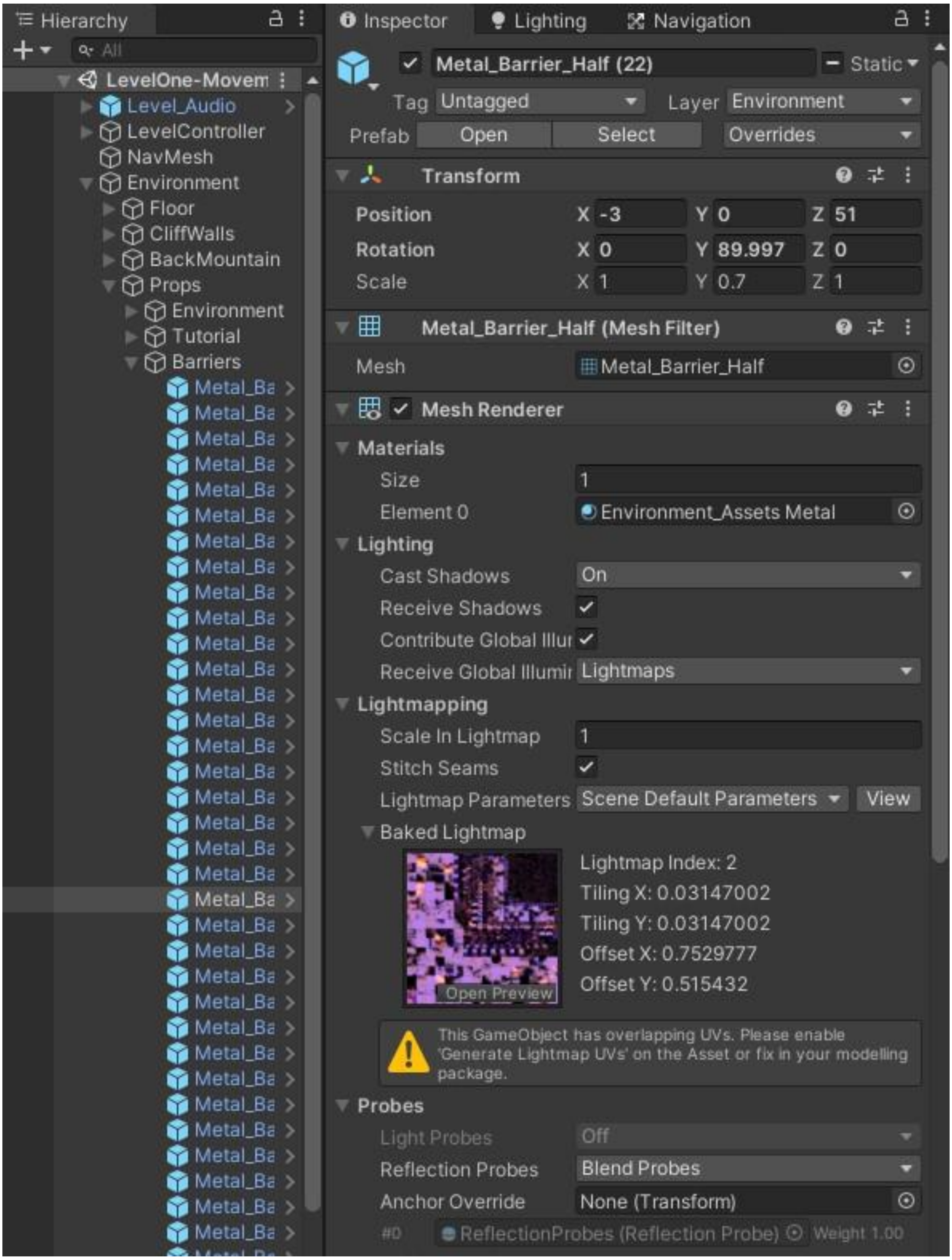
Probes

Light Probes Off

Reflection Probes Blend Probes

Anchor Override None (Transform)

#0 ReflectionProbes (Reflection Probe) Weight 1.00



Metal_Barrier_Half (22)

Tag Untagged
Layer Environment

Transform

Position	X	-3	Y	0	Z	51
Rotation	X	0	Y	89.997	Z	0
Scale	X	1	Y	0.7	Z	1

Metal_Barrier_Half (Mesh Filter)

Mesh Metal_Barrier_Half

Mesh Renderer

Materials

Size 1

Element 0 Environment_Assets Metal

Lighting

Cast Shadows On

Receive Shadows ✓

Contribute Global Illur ✓

Receive Global Illumir Lightmaps

Lightmapping

Scale In Lightmap 1

Stitch Seams ✓

Lightmap Parameters Scene Default Parameters

Baked Lightmap

Lightmap Index: 2
Tiling X: 0.03147002
Tiling Y: 0.03147002
Offset X: 0.7529777
Offset Y: 0.515432

This GameObject has overlapping UVs. Please enable 'Generate Lightmap UVs' on the Asset or fix in your modelling package.

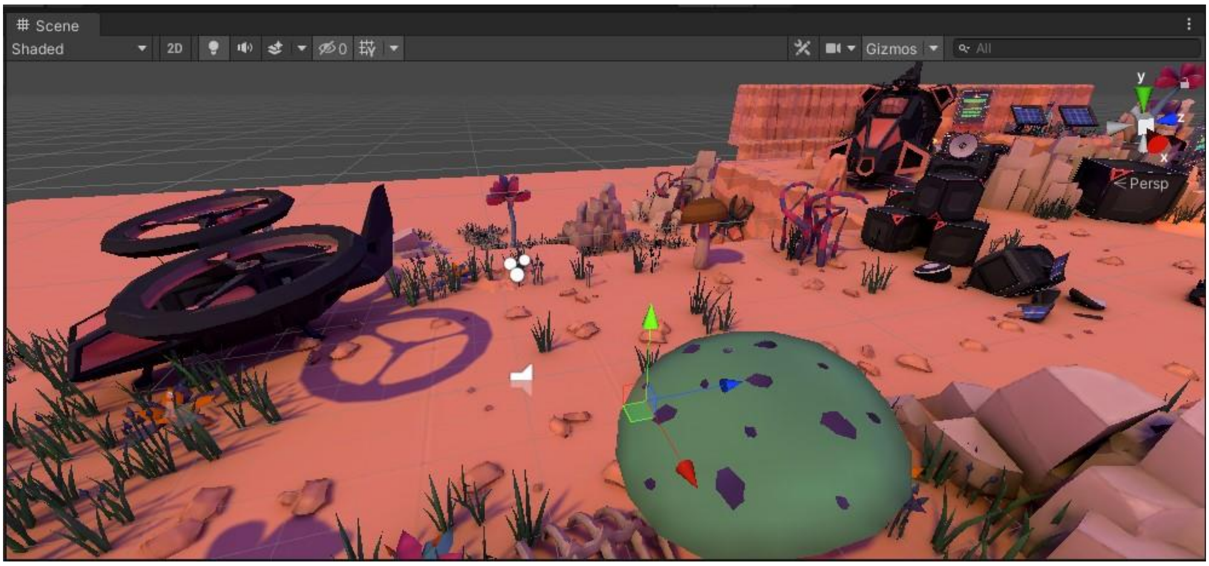
Probes

Light Probes Off

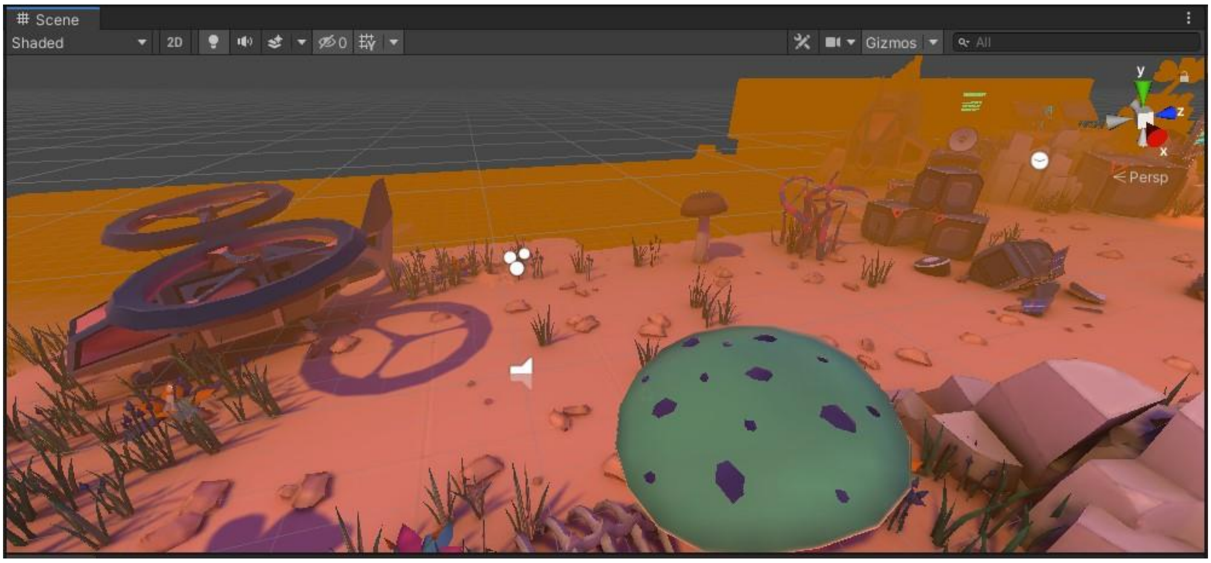
Reflection Probes Blend Probes

Anchor Override None (Transform)

#0 ReflectionProbes (Reflection Probe) Weight 1.00



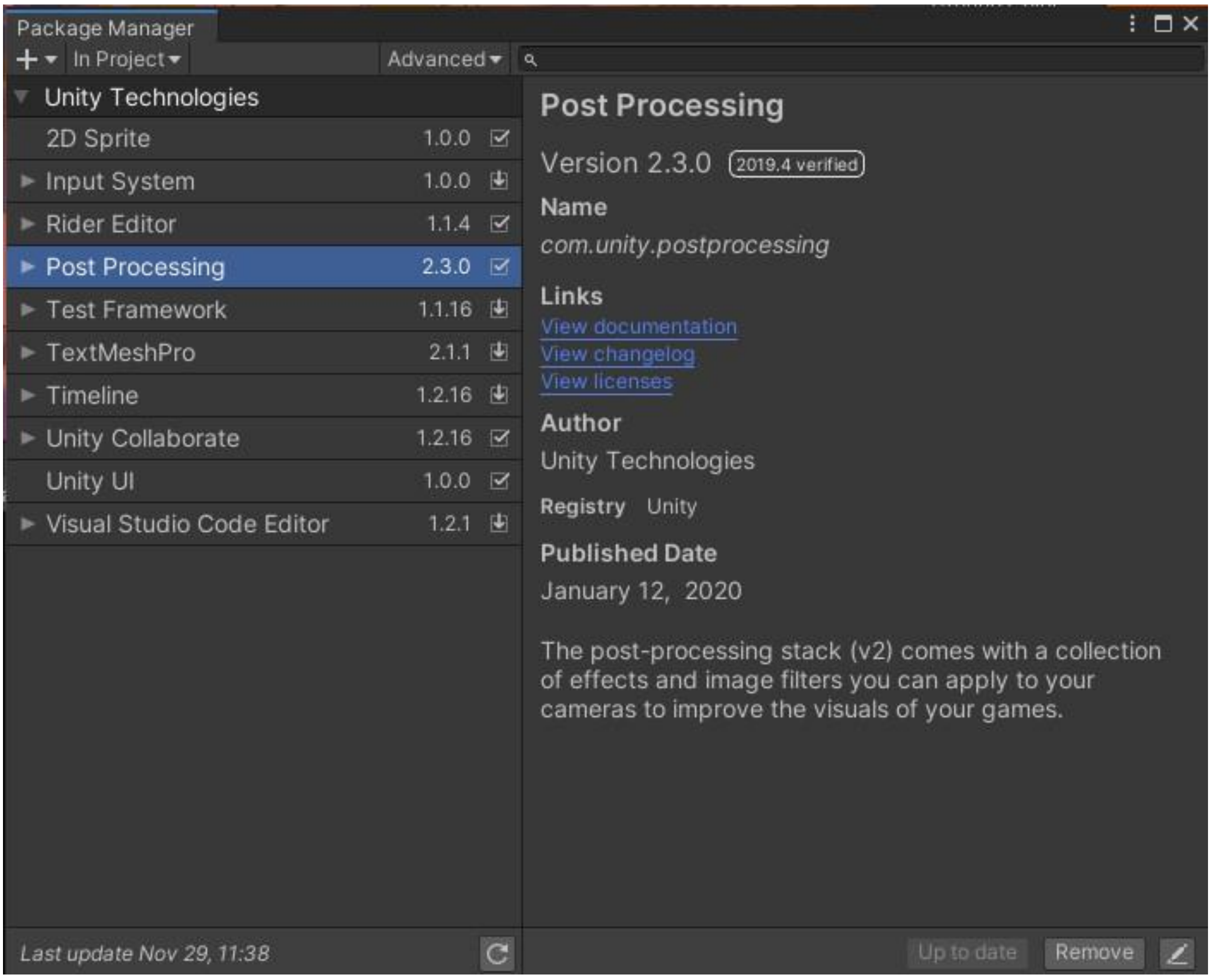
LevelOne-Movement (Before Baking Lights)



LevelOne-Movement (After Baking Lights)

❖ PostProcessing.

- First, we have to make sure the “PostProcessing” plugin in enabled from the “Package Manager”.



Unity Technologies

2D Sprite	1.0.0	☑
Input System	1.0.0	⤓
Rider Editor	1.1.4	☑
Post Processing	2.3.0	☑
Test Framework	1.1.16	⤓
TextMeshPro	2.1.1	⤓
Timeline	1.2.16	⤓
Unity Collaborate	1.2.16	☑
Unity UI	1.0.0	☑
Visual Studio Code Editor	1.2.1	⤓

Last update Nov 29, 11:38

Post Processing

Version 2.3.0 2019.4 verified

Name

com.unity.postprocessing

Links

View documentation

View changelog

View licenses

Author

Unity Technologies

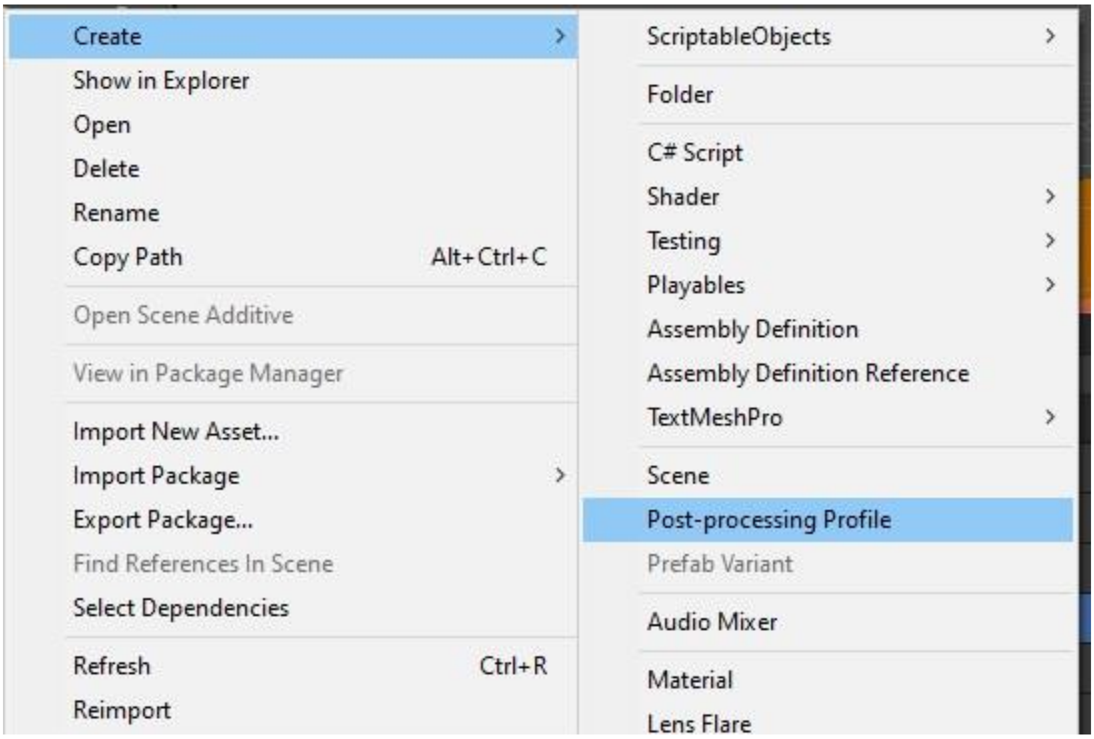
Registry Unity

Published Date

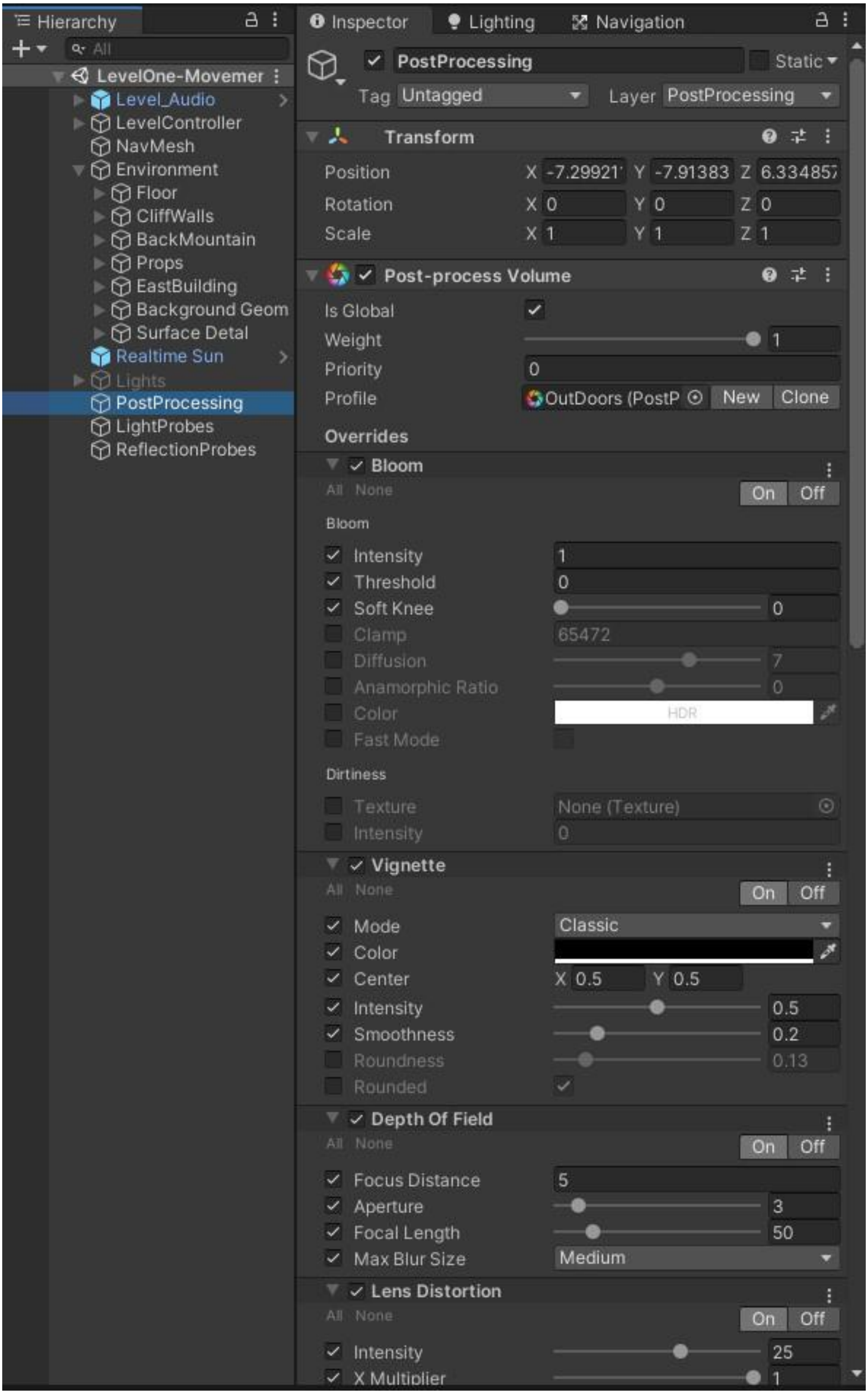
January 12, 2020

The post-processing stack (v2) comes with a collection of effects and image filters you can apply to your cameras to improve the visuals of your games.

- Then we create a new folder, “PostProcess” and add the “Post-Processing Profile” by Right clicking and under Create section.



- Then we add Effects as per our requirement.
- I have added effects like
    - Bloom: which makes Reflective Surfaces or Substances Glow.
    - Vignette: which Makes the 4 Corner Black.
    - Depth of Field: which makes the Player stay in Focus and Blur the rest of the surrounding.
    - Lens Distortion: which makes the Lens Zoom towards the Screen. In my case I have made the Centre bit close to the Screen and the 4 Corners go bit far from the screen.
    - Grain: which adds some noise to the screen, just to make it look realistic.
    - Color Grading: which is responsible for the Color, Temperature and Saturation of the Visual Representation of the Game.



LevelOne-Movemer
Level_Audio
LevelController
NavMesh
Environment
Floor
CliffWalls
BackMountain
Props
EastBuilding
Background Geom
Surface Detal
Realtime Sun
Lights
PostProcessing
LightProbes
ReflectionProbes

☑ PostProcessing

Tag Untagged
Layer PostProcessing

Transform

Position X -7.29921 Y -7.91383 Z 6.334857
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

☑ Post-process Volume

Is Global ☑
Weight 1
Priority 0
Profile OutDoors (PostP

Overrides

☑ Bloom

All None

Bloom

☑ Intensity 1
☑ Threshold 0
☑ Soft Knee 0
□ Clamp 65472
□ Diffusion 7
□ Anamorphic Ratio 0
□ Color
□ Fast Mode

Dirtiness

□ Texture None (Texture)
□ Intensity 0

☑ Vignette

All None

☑ Mode Classic
☑ Color
☑ Center X 0.5 Y 0.5
☑ Intensity 0.5
☑ Smoothness 0.2
□ Roundness 0.13
□ Rounded ✓

☑ Depth Of Field

All None

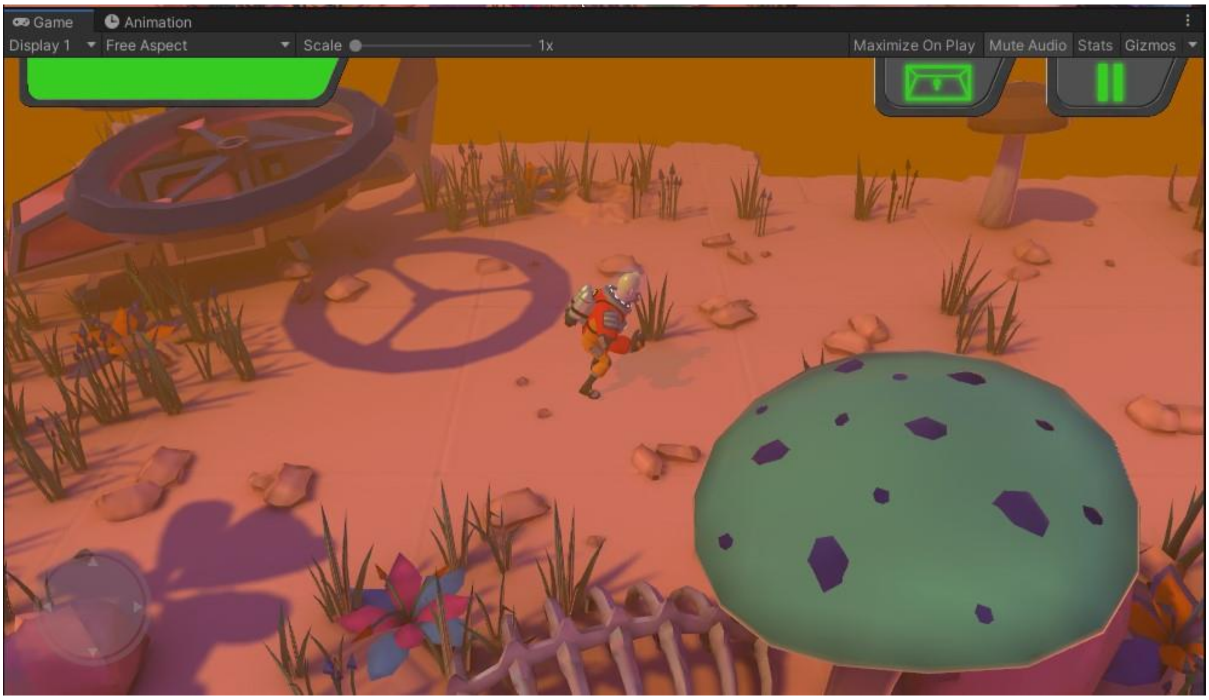
☑ Focus Distance 5
☑ Aperture 3
☑ Focal Length 50
☑ Max Blur Size Medium

☑ Lens Distortion

All None

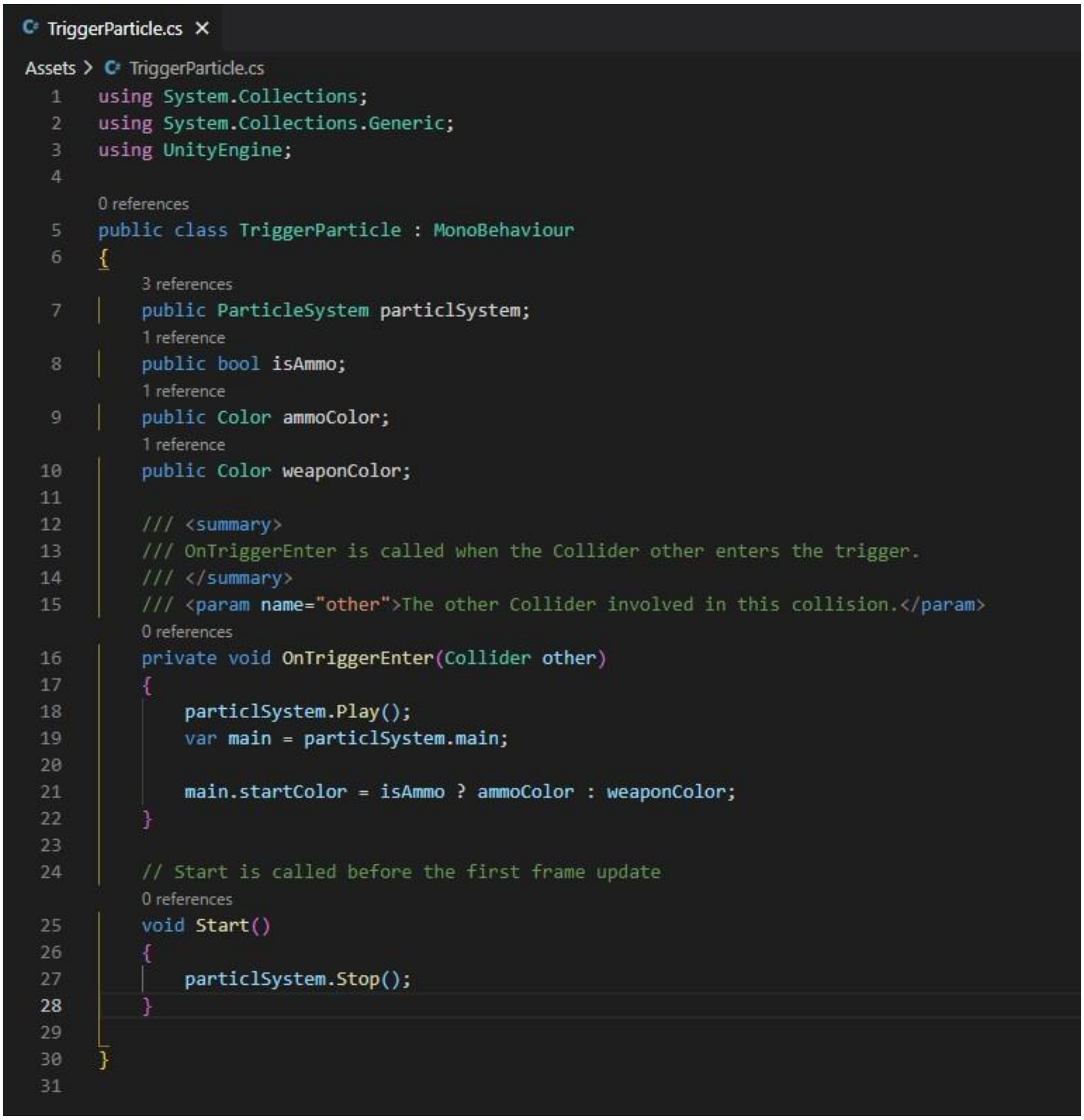
☑ Intensity 25
☑ X Multiplier 1

- And then the Game look like this:

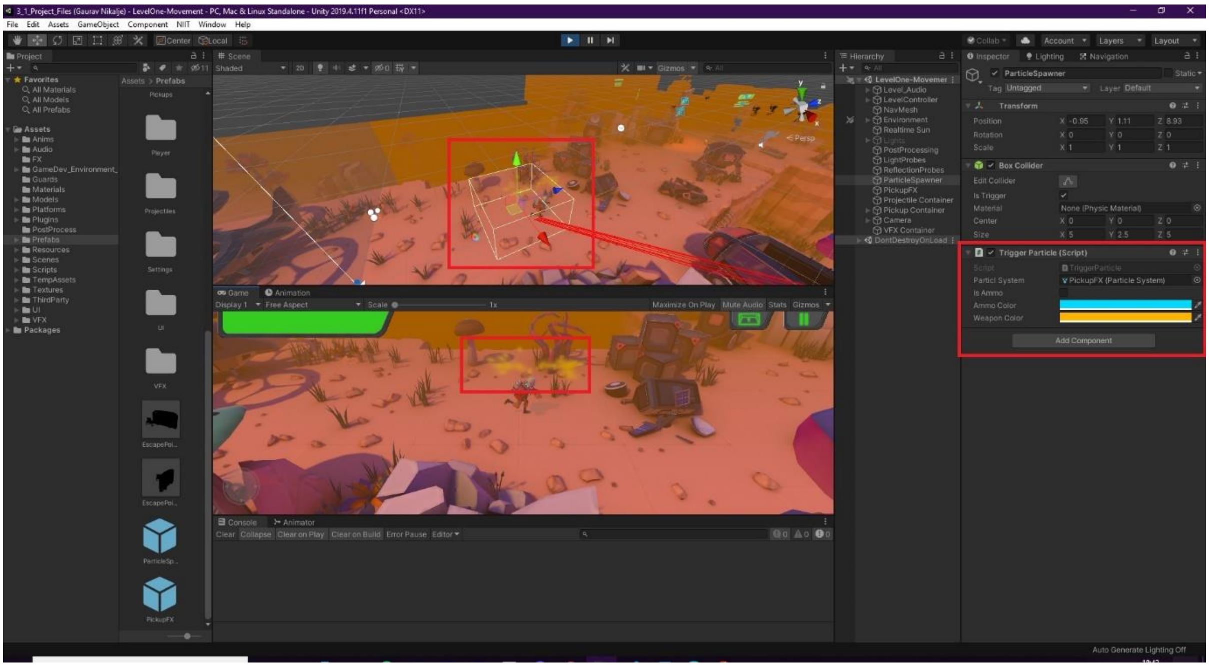


- ❖ **Particle System.**

- We first crate a Particle System and play around the settings and achieve an effect we want.
- Then, we create an Empty GameObject named “PickupSpawner” and to it we add a “Box Collider” and a script.
- And we also make sure that the “isTrigger” check is ticked.
- Then in the script we take in the particle system and set an If-Condition where if the Picked-Up Item is an Ammo, then set the color to some Blue, and if not, then set it to some other color.



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

0 references
public class TriggerParticle : MonoBehaviour
{
    3 references
    public ParticleSystem particlSystem;
    1 reference
    public bool isAmmo;
    1 reference
    public Color ammoColor;
    1 reference
    public Color weaponColor;

    /// <summary>
    /// OnTriggerEnter is called when the Collider other enters the trigger.
    /// </summary>
    /// <param name="other">The other Collider involved in this collision.</param>
    0 references
    private void OnTriggerEnter(Collider other)
    {
        particlSystem.Play();
        var main = particlSystem.main;

        main.startColor = isAmmo ? ammoColor : weaponColor;
    }

    // Start is called before the first frame update
    0 references
    void Start()
    {
        particlSystem.Stop();
    }

}

```



3_1_Project_Files (Gaurav Nikalje) - LevelOne-Movement - PC, Mac & Linux Standalone - Unity 2019.4.11f1 Personal <DX11>

ParticleSpawner Static
Tag Untagged Layer Default

Transform
Position X -0.95 Y 1.11 Z 8.93
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Box Collider
Edit Collider
Is Trigger ✓
Material None (Physic Material)
Center X 0 Y 0 Z 0
Size X 5 Y 2.5 Z 5

Trigger Particle (Script)
Script TriggerParticle
Partic System PickupFX (Particle System)
Is Ammo
Ammo Color
Weapon Color

Step 02: What have I Learned

- Explain the differences between real-time and baked lighting.
  - Real-Time Lighting is a method which projects lighting, Shadows and Ambient Occlusions, real time. That is, if we change the position of any GameObject, the shadow and occlusion changes.
  - While, Baked Lighting is a method where the lighting, Shadows and Ambient Occlusions are already baked, and they do not change their positions when we move a GameObject.
  - Real-Time Lighting is System Expensive, because it calculates the lighting data on every frame, while Baked

Lighting is not System Expensive, as everything is already baked.

- Real-Time needs less settings to be tweaked, while for the Baked Lighting, we might need to set some settings like, LightMap Resolution and its Compression, etc.

- Explain the differences between reflection probes and light probes.

  - Reflection Probes are used for creating reflections on Objects while Light Probes are used to store and create shadows and occlusions.
  - Reflection Probe captures a Spherical View of its surroundings in all directions, while Light Probes provide Higher Quality of Lighting, by bouncing multiple Indirect Lights on all the objects present in the scene.
  - Reflection Probe uses a single Sphere and a Bounding Box, while Light Probe uses multiple Spheres where we place them in respective quantity as per our requirement. We simply duplicate the spheres and place them at a higher quantity in areas where the objects are at high quantity, there we want proper shadows.

-------------------------------THE END----------------------------