GAME PROGRAMMING LAB 6

NAME : OM SUBRATO DEY

REGISTER NO.: 21BAI1876

INSTRUCTIONS AS PER GUIDELINES

LAB – 6

Topics Covered:

1. **Audio – Events/ Background**
2. **Light Effects**

Lab Exercise:

Scene 1:

- Add sound effects for the following actions:
  - Left & right move
  - Jump
  - Collectable
  - Background music

Scene 2:

- Press `CTRL+D` on Scene 1 to duplicate it.
- Ensure the background sound continues playing across Scene 1 and Scene 2 (both back and forth).
- Use **point light**, **spot light**, and **area light** in one of the scenes, placing them to appropriately illuminate objects.

Audio Integration

Steps:

1. **Script Implementation:**
   - Use a script with conditions to trigger audio events.
   - Example steps:
     - Create an `AudioSource` object.
     - Invoke the `Play` function using the `AudioSource` variable for each event.
     - Add these events under the corresponding conditions in the script.

Example Code:

```
csharp
Copy code
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Move : MonoBehaviour
{
    public float speed = 8f;
    public AudioSource movementsound;
    public AudioSource movementsound1;

    // Update is called once per frame
    void Update()
    {
        if (Input.GetKey(KeyCode.RightArrow))
        {
            transform.position += new Vector3(speed * Time.deltaTime,
0.0f, 0.0f);
            movementsound.Play();
        }

        if (Input.GetKey(KeyCode.LeftArrow))
        {
            transform.position -= new Vector3(speed * Time.deltaTime,
0.0f, 0.0f);
            movementsound1.Play();
        }
    }
}
```

2. **Download Resources:**
   - **Universe Sound Free Pack:** Universe Sounds Free Pack | Sci-Fi Ambient | Unity Asset Store
   - **Sound FX – Retro pack:** Retro Games Sound FX | Audio Sound FX | Unity Asset Store
3. **Player Inspector Setup:**
   - Add the script to the player.
   - Go to **Component > Audio Source**, choose an audio clip, and disable `Play on Awake` to trigger audio only on event conditions.
4. **Background Music Setup:**
   - Add the following script to the main camera to ensure background music plays throughout the game:

```
csharp
Copy code
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class bgmusic : MonoBehaviour
{
    public AudioSource bgm;

    void Start()
```

```
    {
        bgm = GetComponent<AudioSource>();
        bgm.Play();
    }
}
```

5. **Load Level Script:** (Used to load new scenes at specific trigger points)

```
csharp
Copy code
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class LoadLevel : MonoBehaviour
{
    public string level_name;

    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.gameObject.name == "Player")
        {
            SceneManager.LoadScene(level_name);
        }
    }
}
```

Supporting video link:
https://www.youtube.com/watch?v=J77CMuAwVDY&ab_channel=CodinginFlow

---

Light Effects

Spot & Point Light Overview:

- **Spot Lights:** Ideal for flashlights, car headlights, and searchlights. Use for focused lighting in small areas.
- **Point Lights:** Emit light equally in all directions from a point, typically used for light bulbs.

Area Light Setup:

Steps to Visualize Area Light in Unity:

1. **Create a Scene:**
   - Either create a new scene or use an existing one.
2. **Set Up Basic Scene:**
   - Add static objects (e.g., cubes, spheres) to observe the Area Light effect.
3. **Create an Area Light:**
   - Right-click in the **Hierarchy** window → **Light > Area Light**.
4. **Configure Area Light:**
   - Change **Light Mode** to `Baked`.
   - Adjust **Width** and **Height** to fit the light source (e.g., window size).
   - Set the **Intensity** and **Color** for desired lighting effects.
5. **Set Scene for Baked Lighting:**
   - Go to **Window > Rendering > Lighting**.
   - Enable `Auto Generate` or manually click **Generate Lighting** after setting up the scene.
6. **Bake Lighting:**
   - Ensure objects are set to `Static` and bake the lighting.
7. **Visualize and Adjust:**
   - After baking, observe the soft shadows and diffused light from the Area Light. Adjust **Width**, **Height**, **Range**, and **Intensity** as needed.
8. **Test Different Materials:**
   - Apply different materials to scene objects to observe how they react to the Area Light.

Sample Scenes & Area Light Positioning:

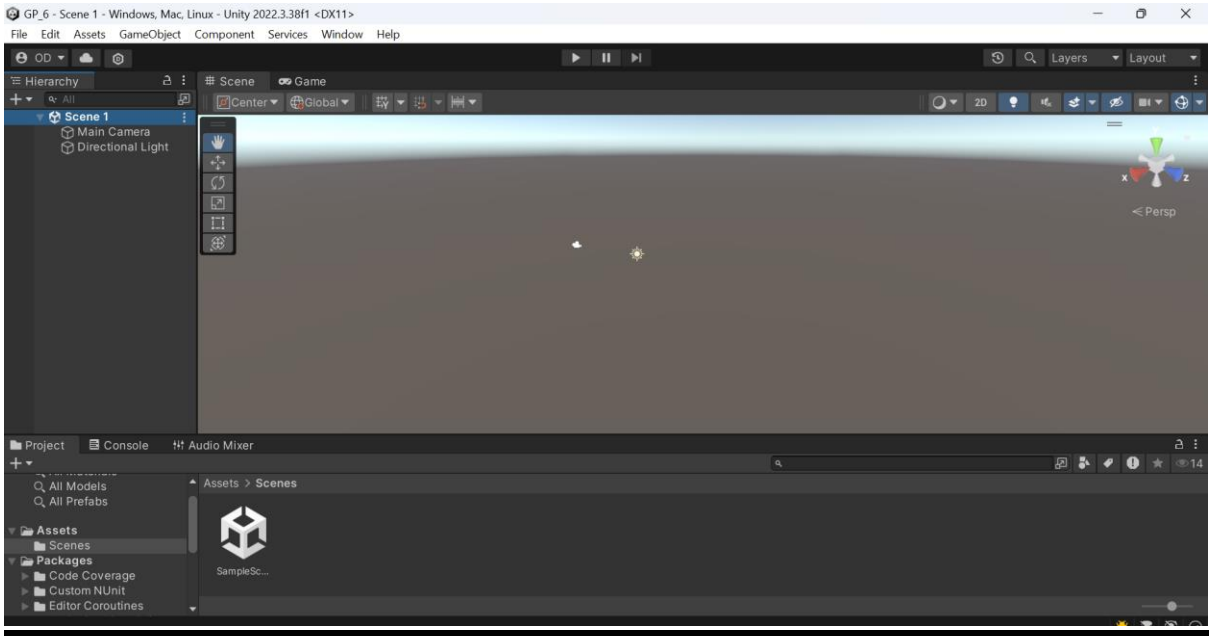
- Experiment with placing multiple area lights to achieve mixed lighting effects.

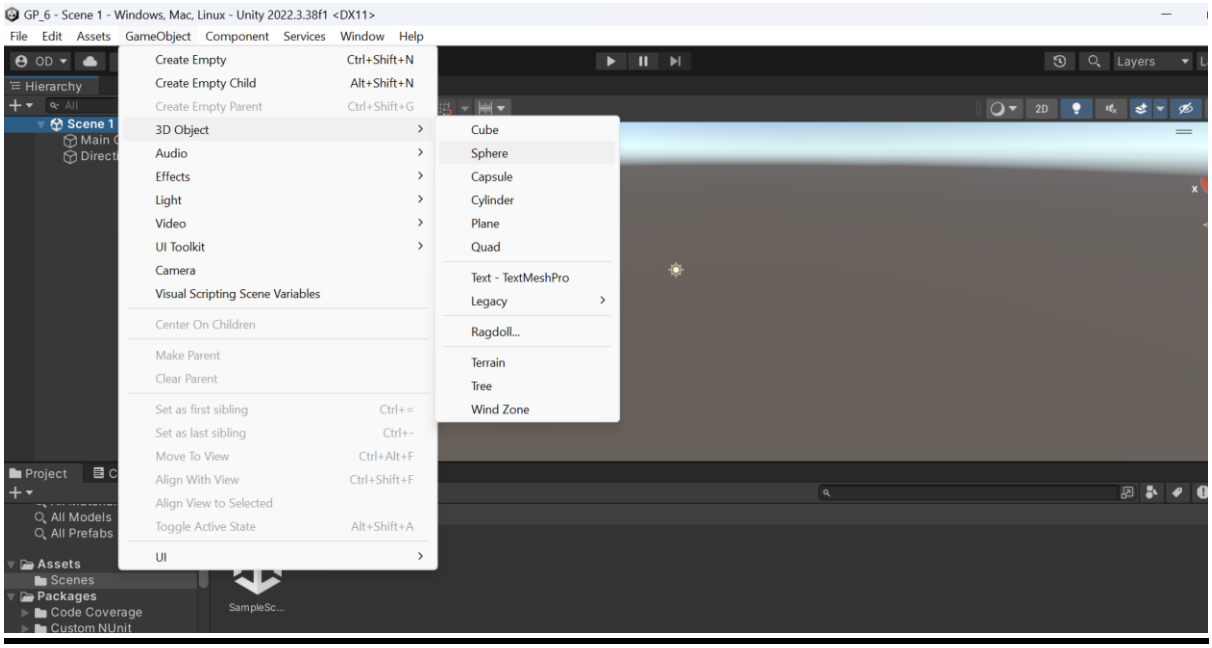
For additional details on Unity's lighting features, visit the

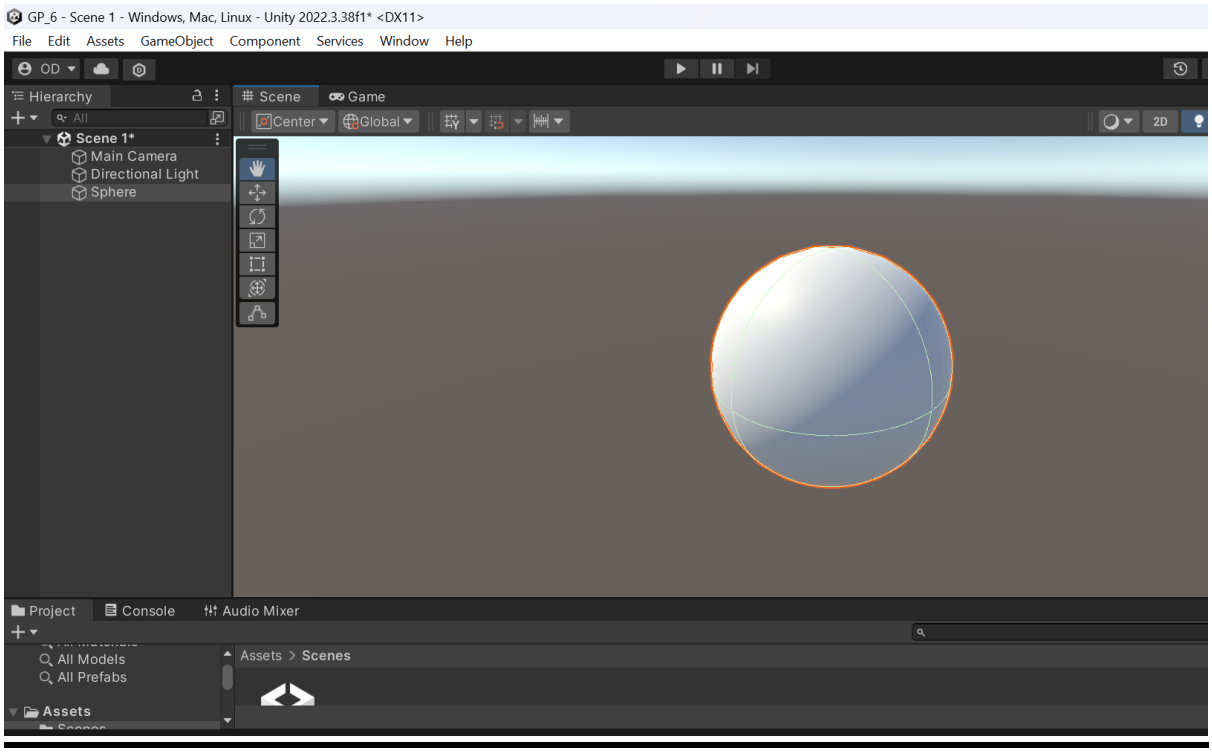
➔ https://docs.unity3d.com/560/Documentation/Manual/Lighting.html#:~:text=Like%20a%20point%20light%2C%20a,direction%20of%20the%20light%20object.

AUDIO EFFECTS PART

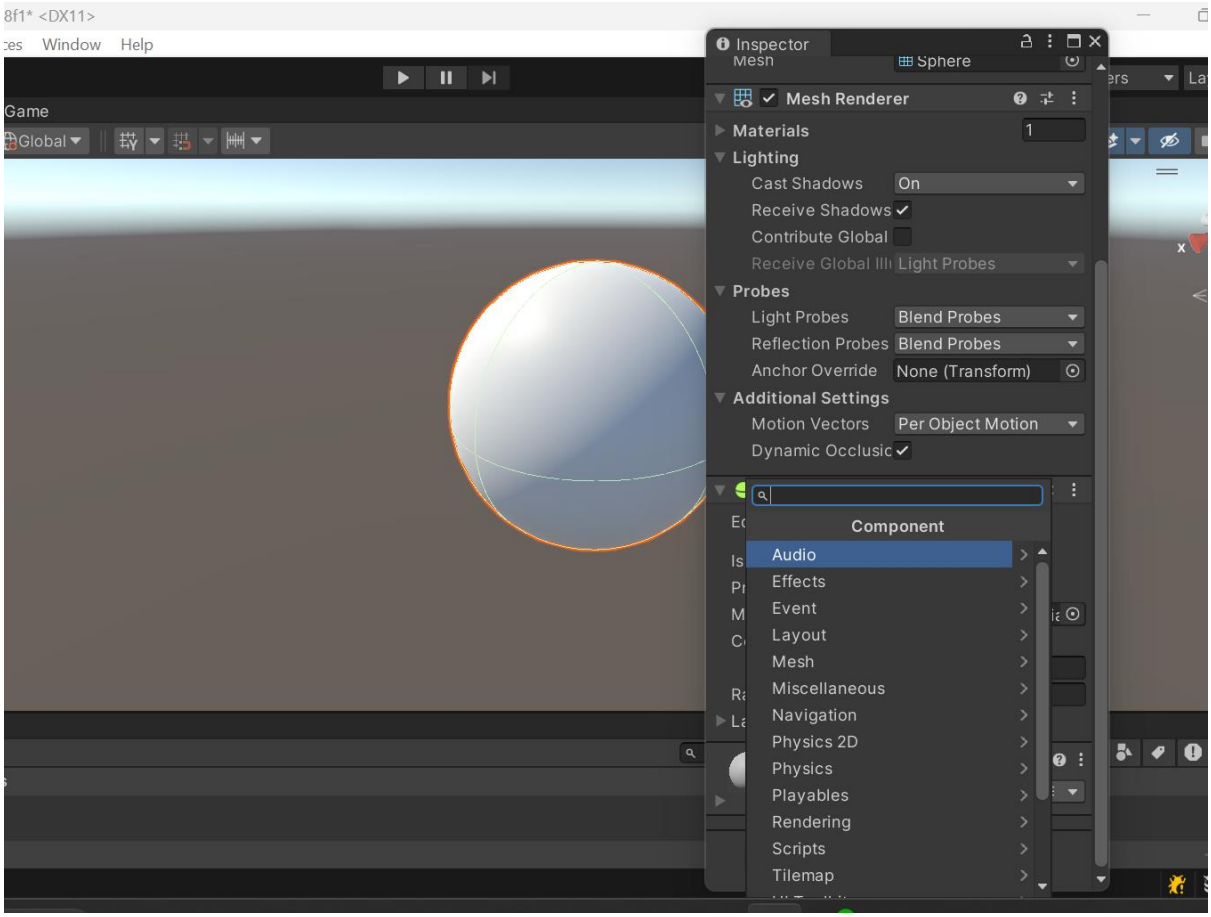
Step 1: Creating object for scene 1 with required specifications here.

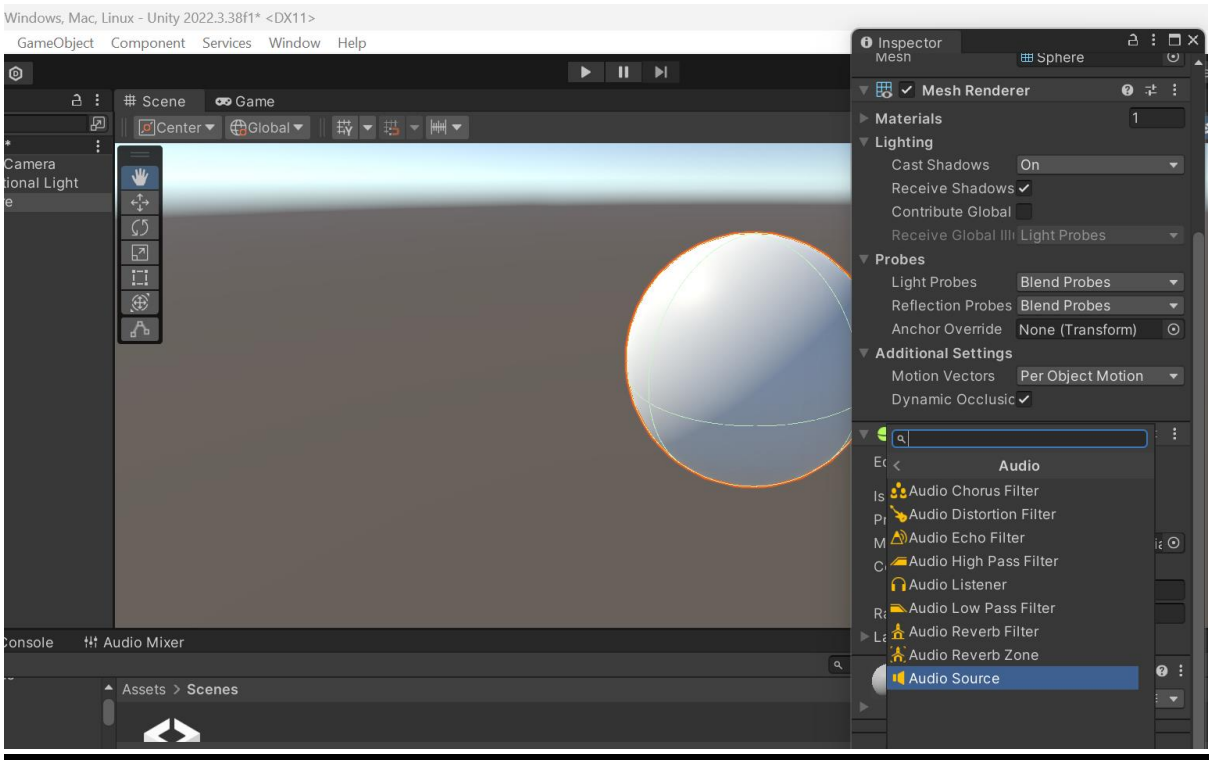


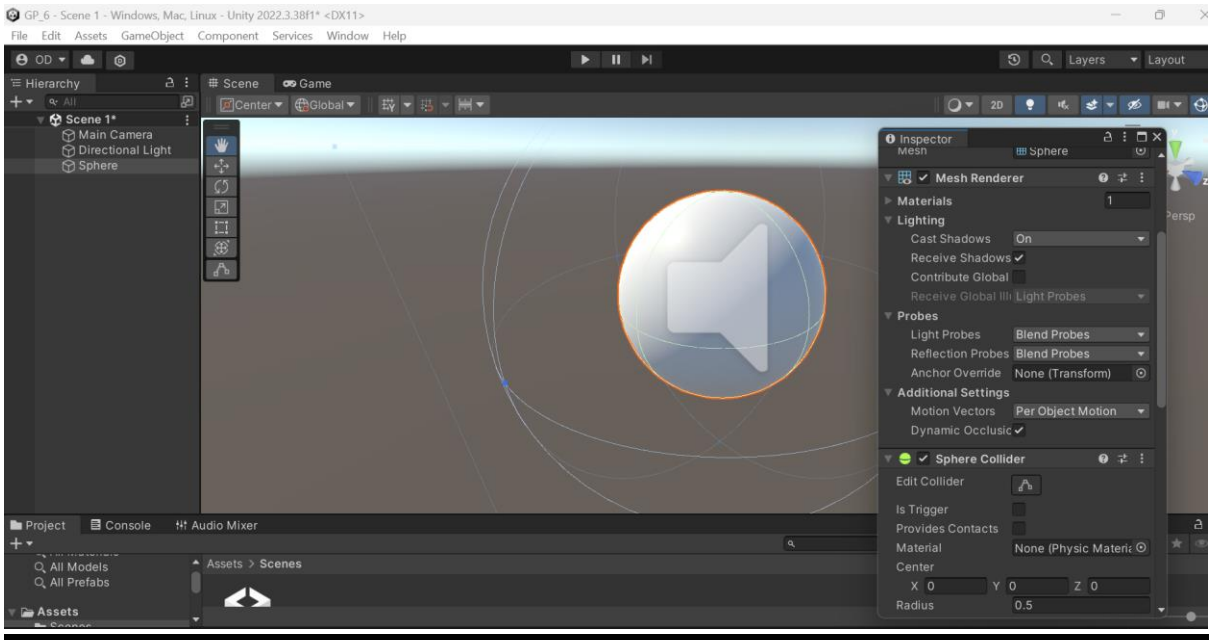




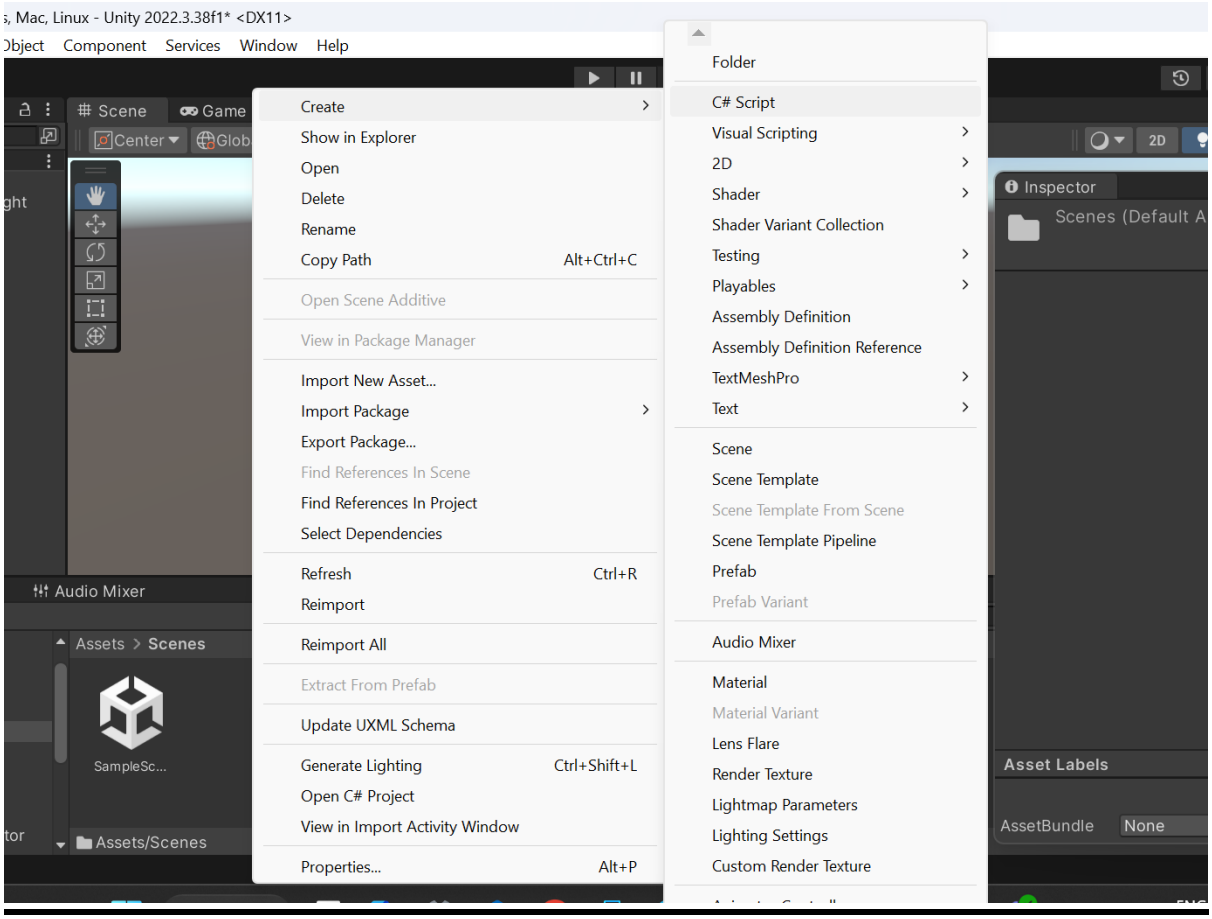
Step 2: Adding audio effects for the object in the scene

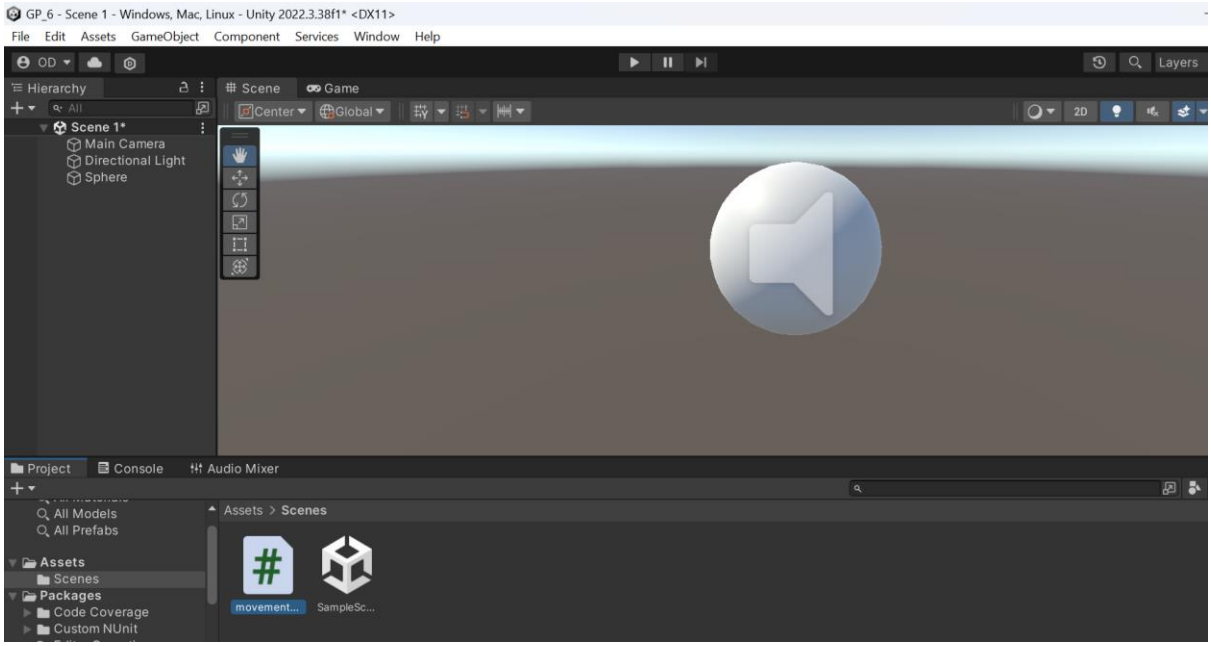

Inspector
Sphere
Mesh Renderer
Materials
Lighting
Cast Shadows
On
Receive Shadows
Contribute Global
Probes
Light Probes
Blend Probes
Reflection Probes
Blend Probes
Anchor Override
None (Transform)
Additional Settings
Motion Vectors
Per Object Motion
Component
Audio
Effects
Event
Layout
Mesh
Miscellaneous
Navigation
Physics 2D
Physics
Playables
Rendering
Scripts
Tilemap


GameObject
Component
Services
Window
Help
Scene
Game
Inspector
Sphere
Mesh Renderer
Materials
Lighting
Cast Shadows
On
Receive Shadows
Contribute Global
Probes
Light Probes
Blend Probes
Reflection Probes
Blend Probes
Anchor Override
None (Transform)
Additional Settings
Motion Vectors
Per Object Motion
Audio
Audio Chorus Filter
Audio Distortion Filter
Audio Echo Filter
Audio High Pass Filter
Audio Listener
Audio Low Pass Filter
Audio Reverb Filter
Audio Reverb Zone
Audio Source
Audio Mixer
Assets
Scenes

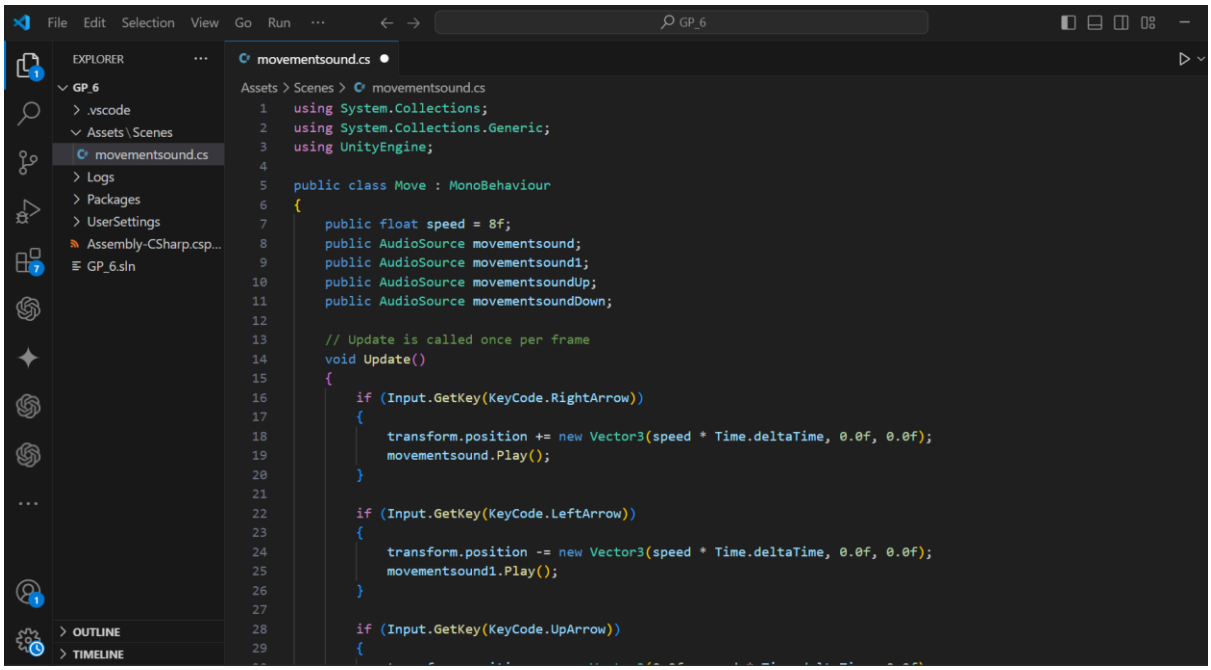


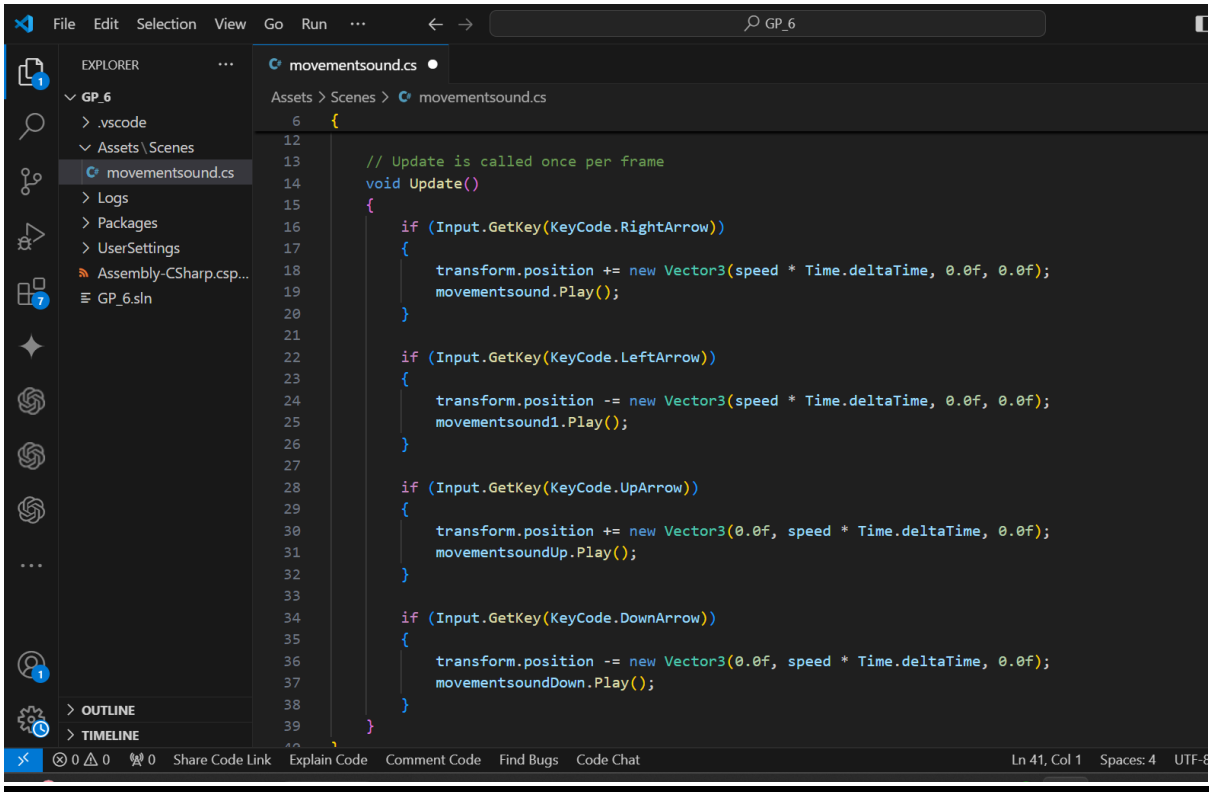
Step 3: Adding the movement-sound script.



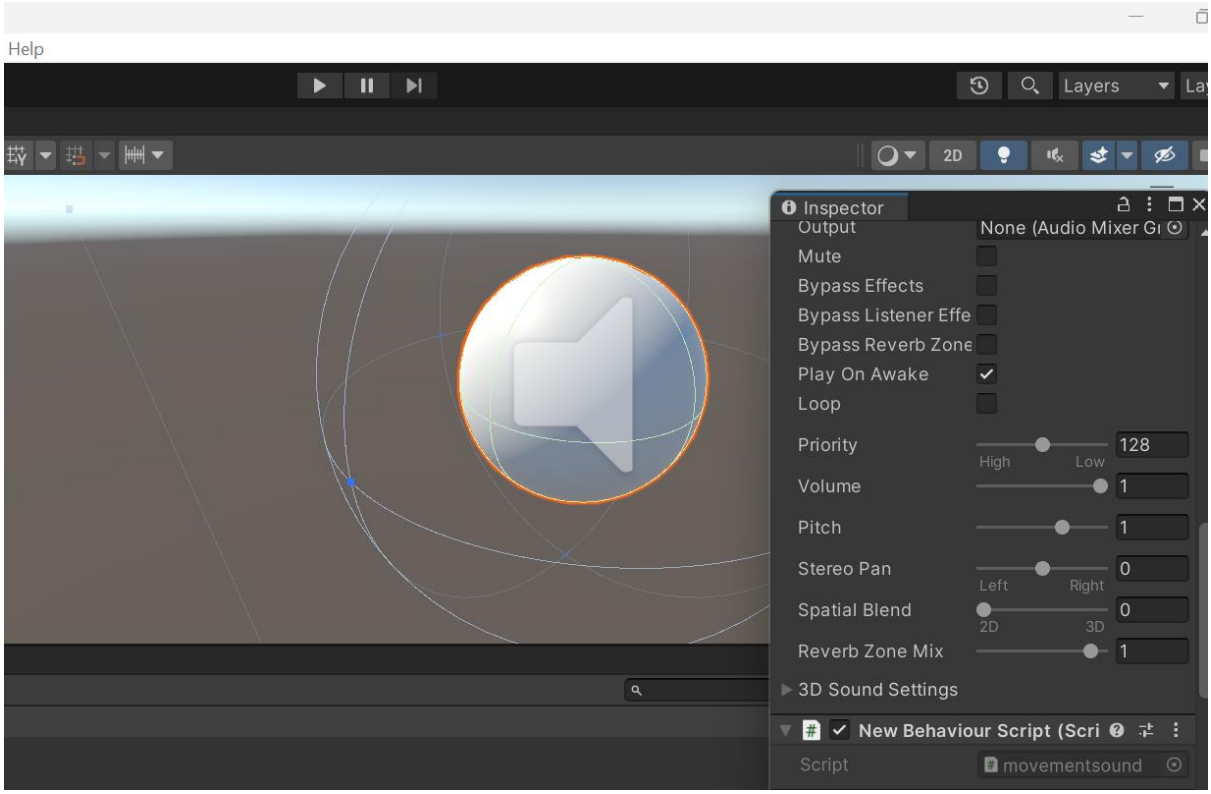


Step 4: Modifying the C# Script

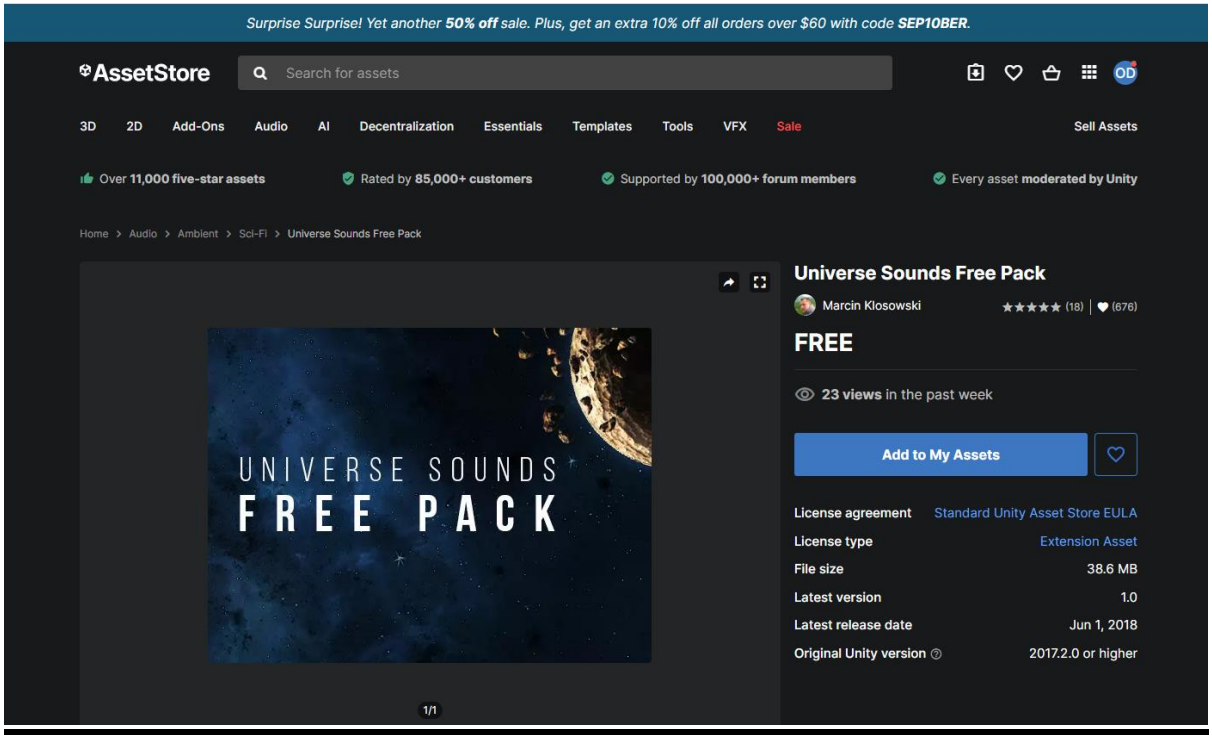


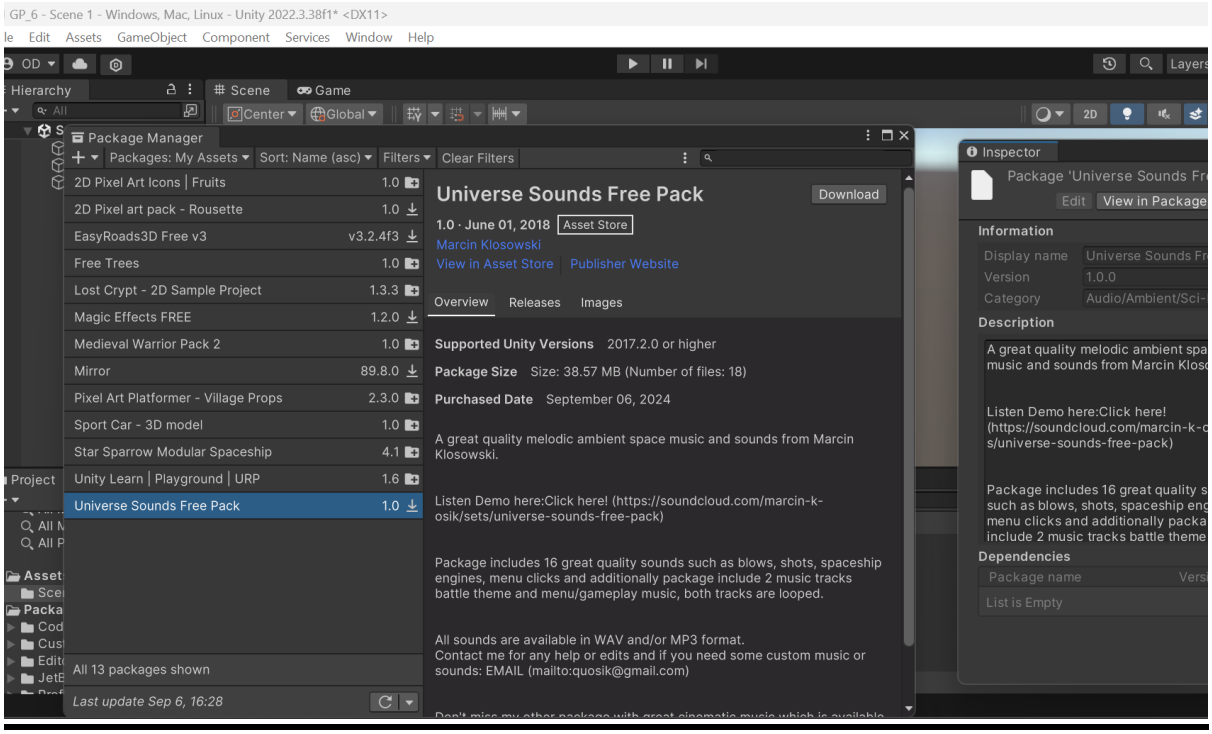


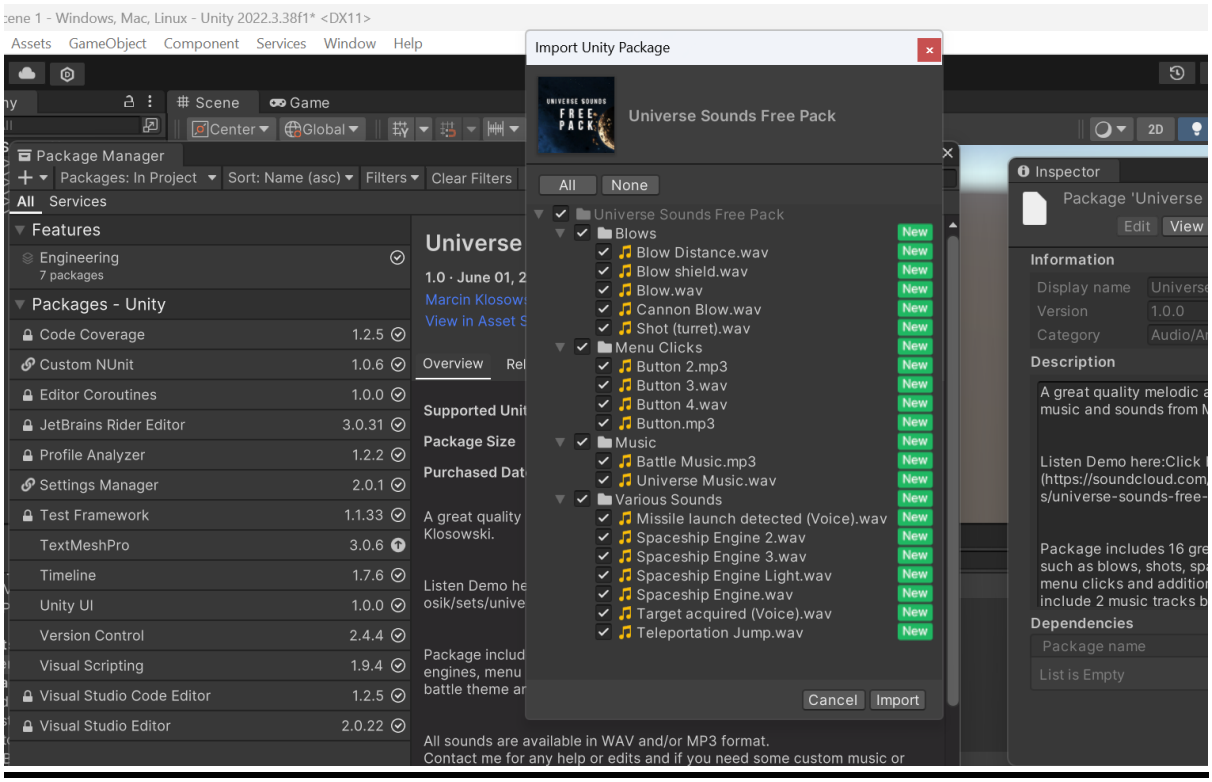
Step 5: Adding above script as component of the sphere.

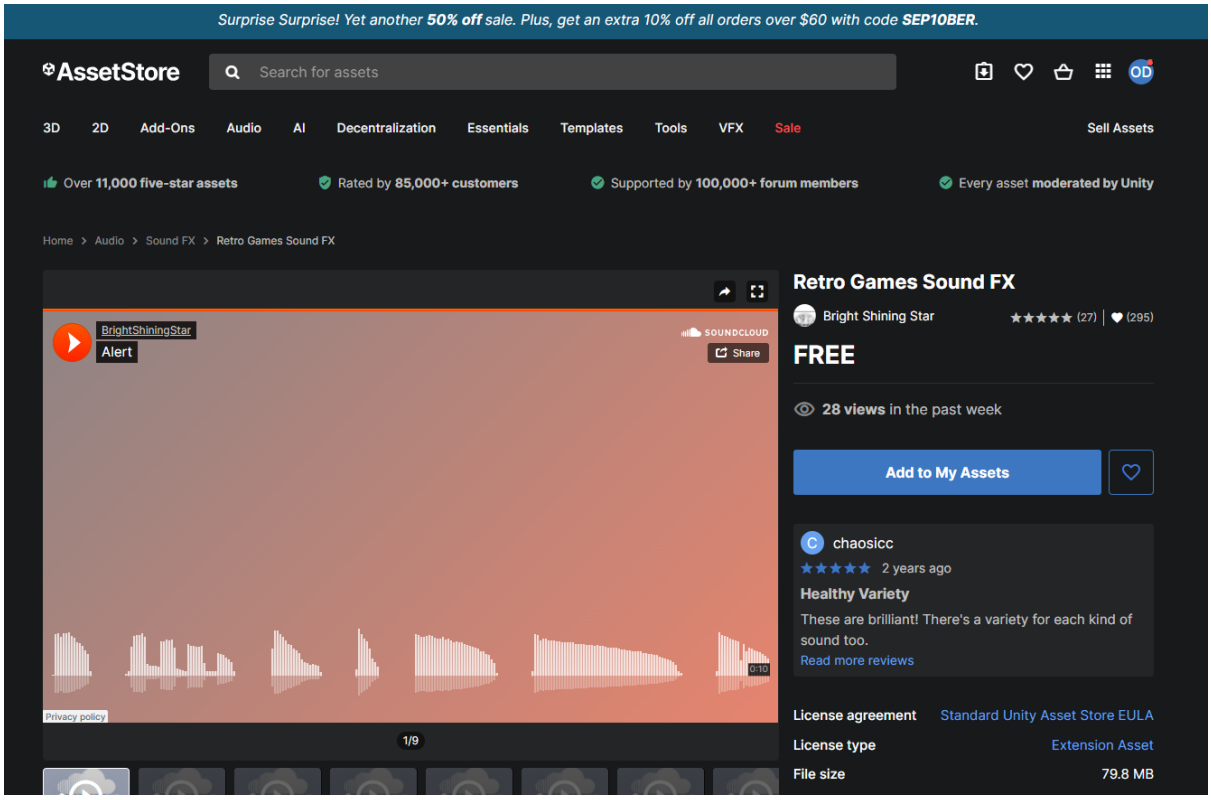


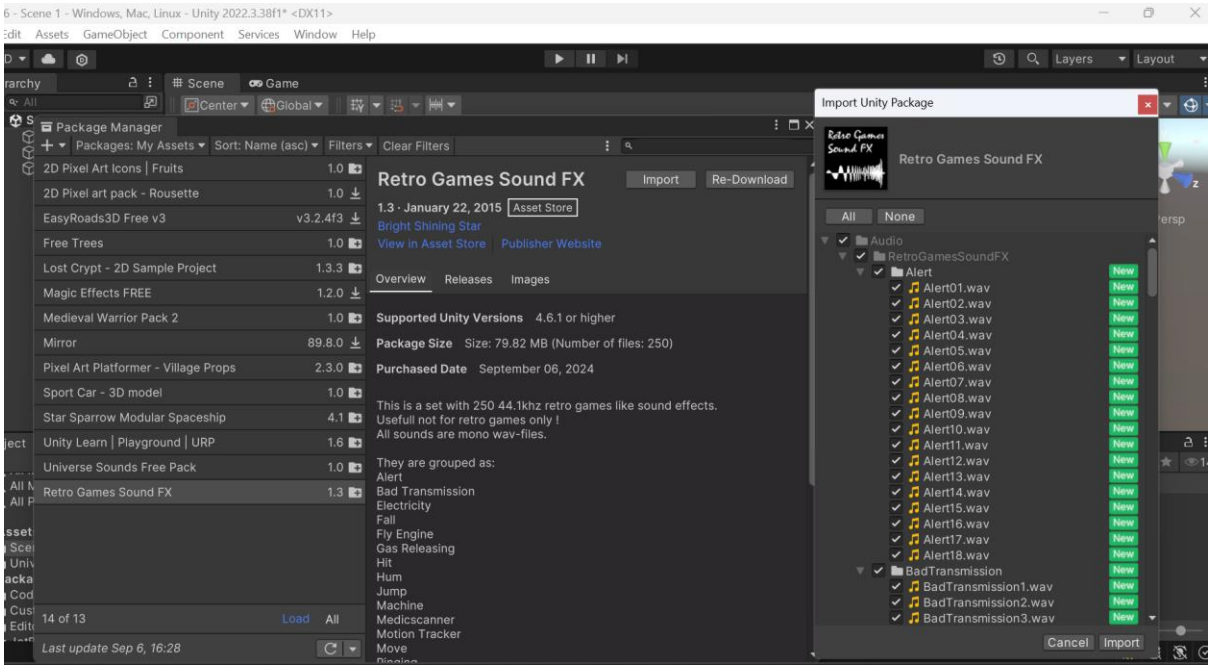
Step 6: Installing the necessary Universe Sound and Sound – FX Retro pack assets from AssetStore Unity and importing elements.



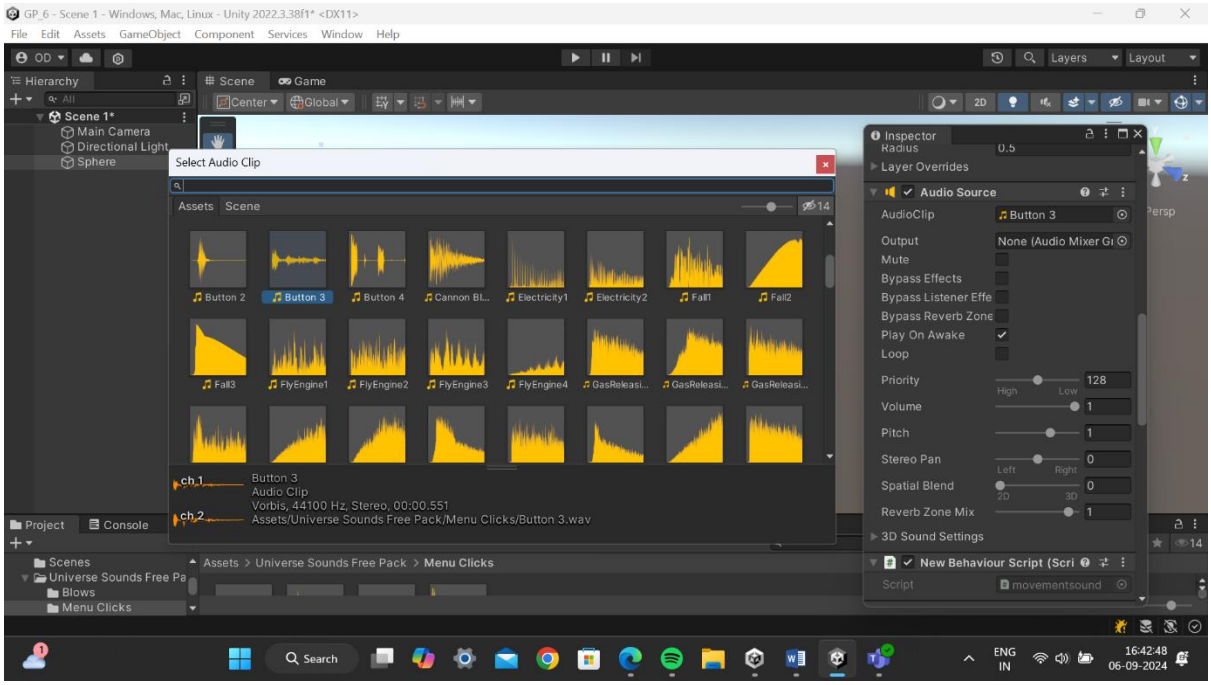

GP_6 - Scene 1 - Windows, Mac, Linux - Unity 2022.3.38f1* <DX11>
Package Manager
Universe Sounds Free Pack
Download
Inspector
Overview
Releases
Images
All 13 packages shown


Import Unity Package
Universe Sounds Free Pack
All
None
Cancel
Import
Package Manager
Inspector

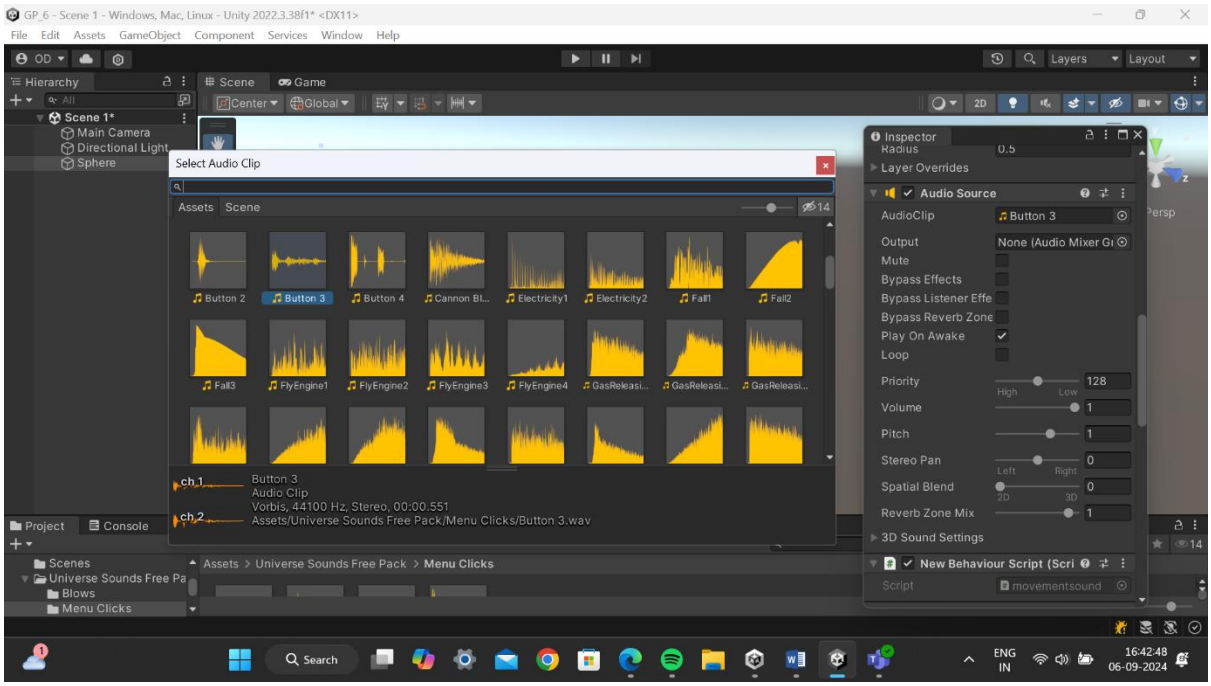

Surprise Surprise! Yet another 50% off sale. Plus, get an extra 10% off all orders over $60 with code SEP10BER.
AssetStore
Search for assets
3D 2D Add-Ons Audio AI Decentralization Essentials Templates Tools VFX Sale Sell Assets
Over 11,000 five-star assets
Rated by 85,000+ customers
Supported by 100,000+ forum members
Every asset moderated by Unity
Home > Audio > Sound FX > Retro Games Sound FX
BrightShiningStar
Alert
SOUNDCLOUD
Share
Privacy policy
1/9
Retro Games Sound FX
Bright Shining Star
(27) | (295)
FREE
28 views in the past week
Add to My Assets
chaosicc
2 years ago
Healthy Variety
These are brilliant! There's a variety for each kind of sound too.
Read more reviews
License agreement Standard Unity Asset Store EULA
License type Extension Asset
File size 79.8 MB


Package Manager
Packages: My Assets
Sort: Name (asc)
Filters
Clear Filters
2D Pixel Art Icons | Fruits 1.0
2D Pixel art pack - Rousette 1.0
EasyRoads3D Free v3 v3.2.4f3
Free Trees 1.0
Lost Crypt - 2D Sample Project 1.3.3
Magic Effects FREE 1.2.0
Medieval Warrior Pack 2 1.0
Mirror 89.8.0
Pixel Art Platformer - Village Props 2.3.0
Sport Car - 3D model 1.0
Star Sparrow Modular Spaceship 4.1
Unity Learn | Playground | URP 1.6
Universe Sounds Free Pack 1.0
Retro Games Sound FX 1.3
14 of 13
Load All
Last update Sep 6, 16:28
Retro Games Sound FX
Import
Re-Download
1.3 · January 22, 2015 Asset Store
Bright Shining Star
View in Asset Store Publisher Website
Overview Releases Images
Supported Unity Versions 4.6.1 or higher
Package Size Size: 79.82 MB (Number of files: 250)
Purchased Date September 06, 2024
This is a set with 250 44.1khz retro games like sound effects.
Usefull not for retro games only !
All sounds are mono wav-files.
They are grouped as:
Alert
Bad Transmission
Electricity
Fall
Fly Engine
Gas Releasing
Hit
Hum
Jump
Machine
Mediscanner
Motion Tracker
Move
Import Unity Package
Retro Games Sound FX
All None
Audio
RetroGamesSoundFX
Alert
Alert01.wav
Alert02.wav
Alert03.wav
Alert04.wav
Alert05.wav
Alert06.wav
Alert07.wav
Alert08.wav
Alert09.wav
Alert10.wav
Alert11.wav
Alert12.wav
Alert13.wav
Alert14.wav
Alert15.wav
Alert16.wav
Alert17.wav
Alert18.wav
BadTransmission
BadTransmission1.wav
BadTransmission2.wav
BadTransmission3.wav
Cancel
Import

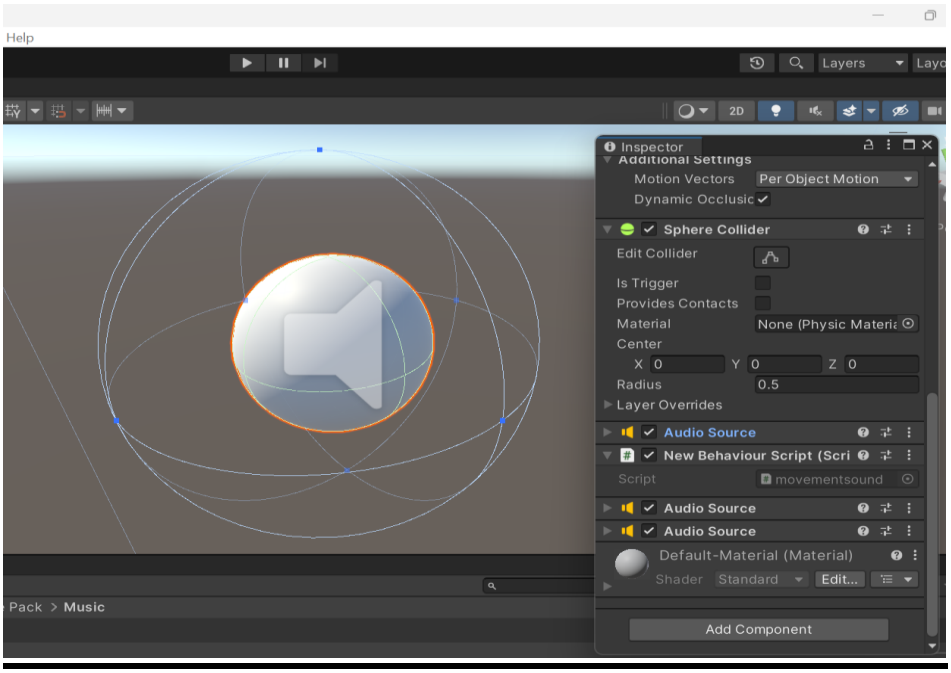
Step 7: Select audio of own choice.



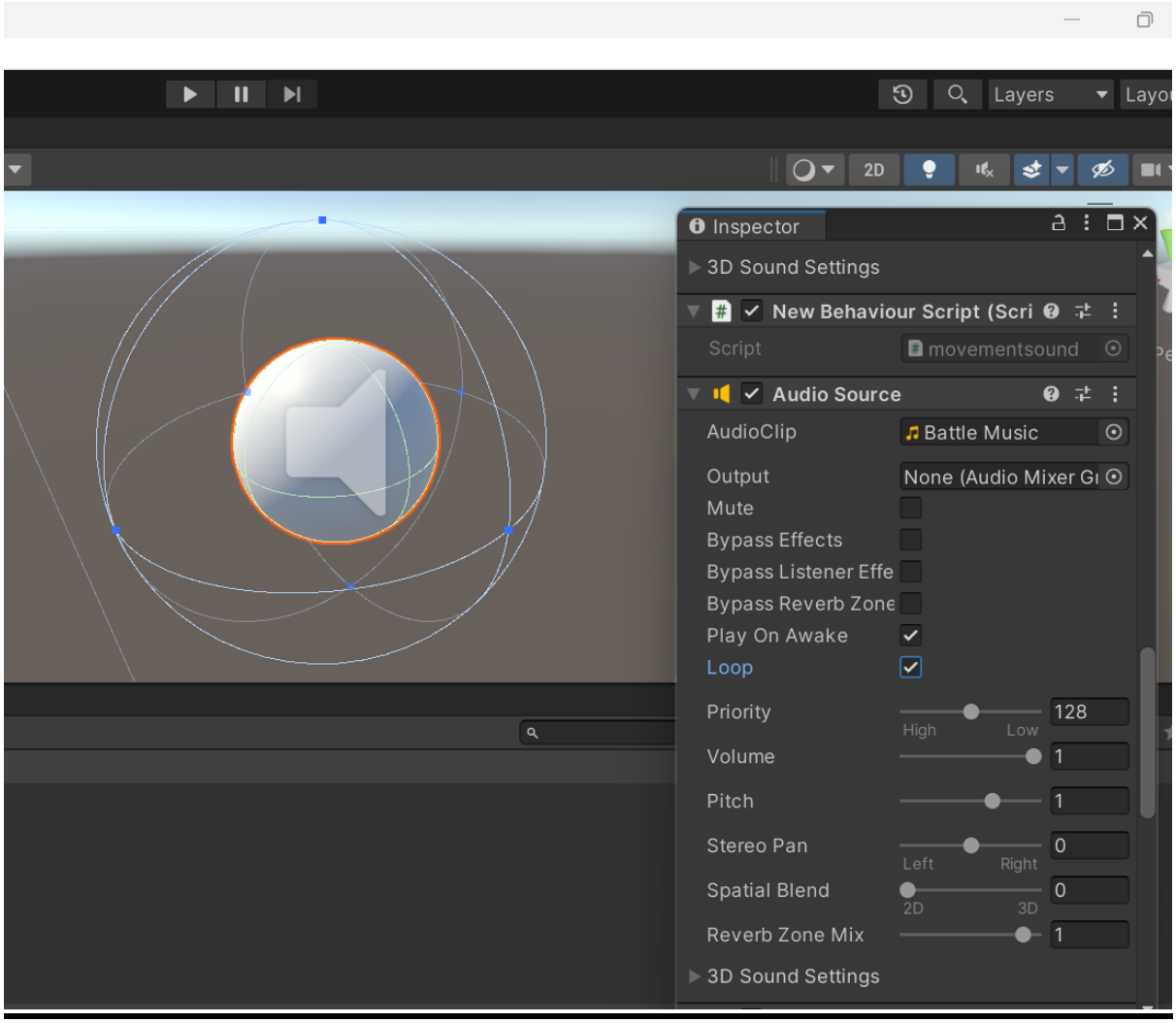
Step 8: Deselecting play on awake for one of the audio sources related to up motion of object in scene.



Step 9: Adding multiple audio sources.

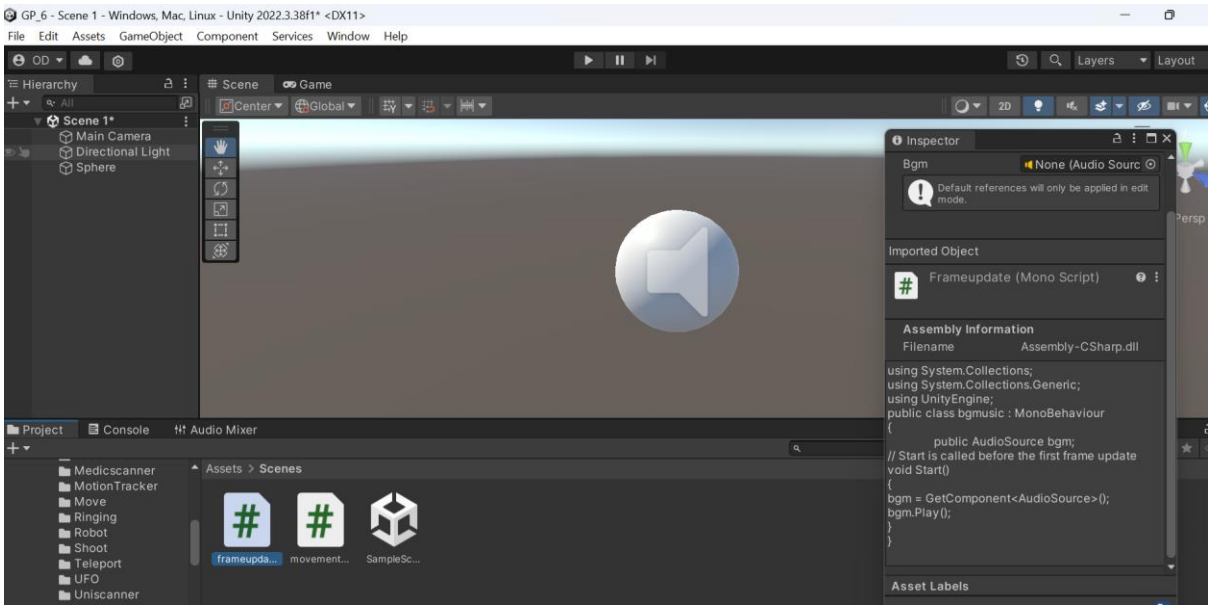


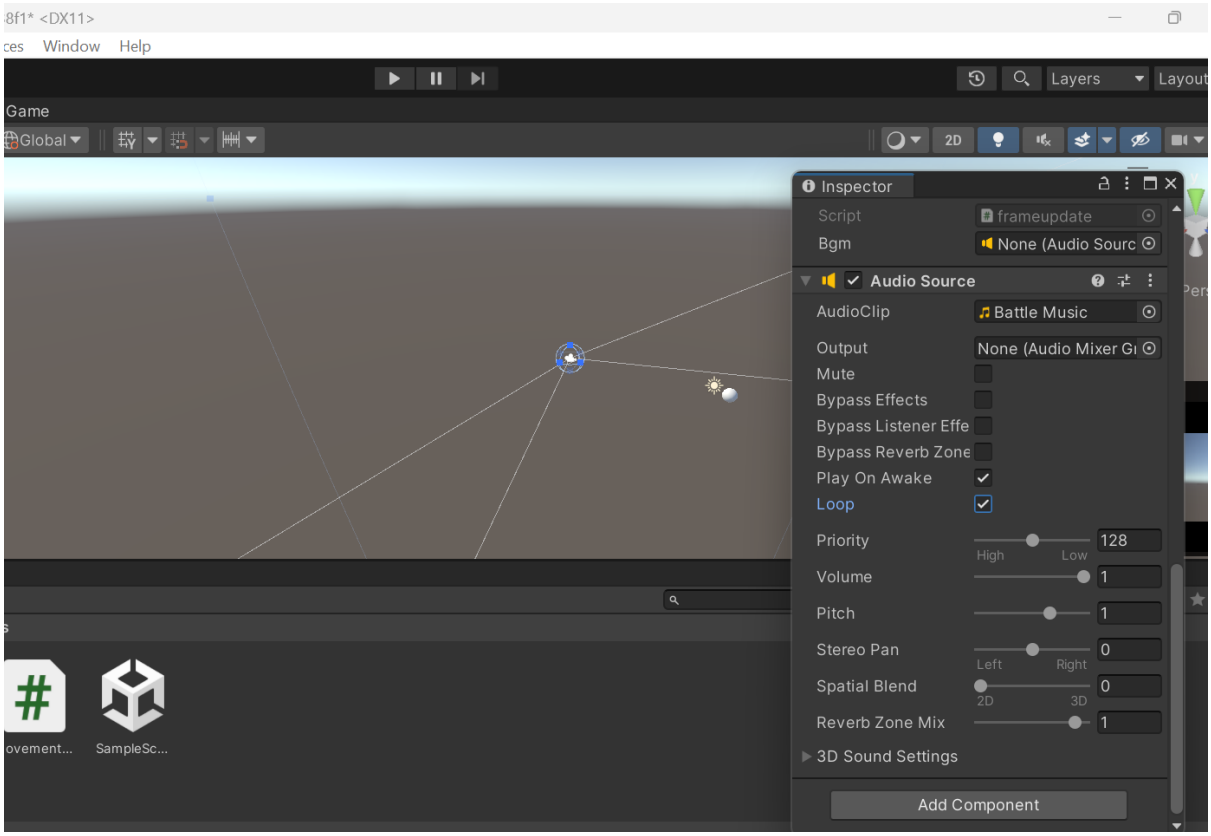
Step 10: Adding tune for background further



Step 11: Adding the following code as script to the main camera so it is available all the time.

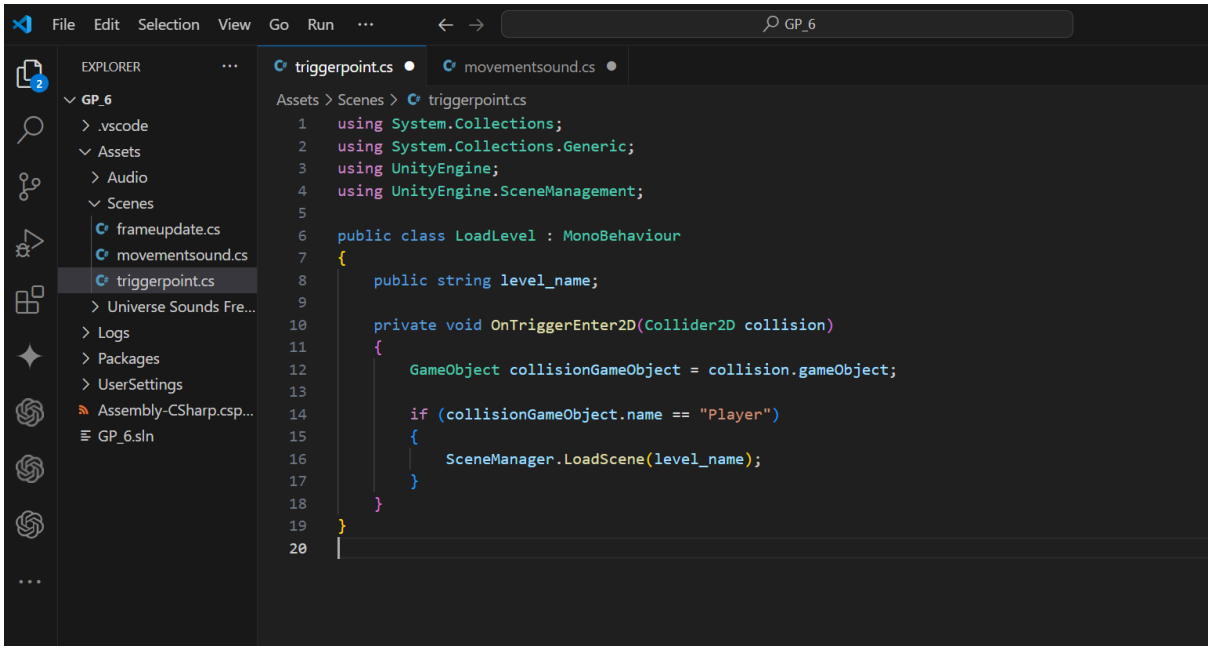
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class bgmusic : MonoBehaviour
{
    public AudioSource bgm;
    // Start is called before the first frame update
    void Start()
    {
        bgm = GetComponent<AudioSource>();
        bgm.Play();
    }
}
```


GP_6 - Scene 1 - Windows, Mac, Linux - Unity 2022.3.38f1* <DX11>
File Edit Assets GameObject Component Services Window Help
Hierarchy
Scene 1*
Main Camera
Directional Light
Sphere
Scene
Game
Inspector
Bgm
None (Audio Sourc
Default references will only be applied in edit mode.
Imported Object
Frameupdate (Mono Script)
Assembly Information
Filename
Assembly-CSharp.dll
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class bgmusic : MonoBehaviour
{
public AudioSource bgm;
// Start is called before the first frame update
void Start()
{
bgm = GetComponent<AudioSource>();
bgm.Play();
}
}
Asset Labels
Project
Console
Audio Mixer
Assets > Scenes
Medicscanner
MotionTracker
Move
Ringing
Robot
Shoot
Teleport
UFO
Uniscanner
frameupda...
movement...
SampleSc...


ces Window Help
Layers
Layout
Game
Global
2D
Inspector
Script
frameupdate
Bgm
None (Audio Sourc
Audio Source
AudioClip
Battle Music
Output
None (Audio Mixer Gr
Mute
Bypass Effects
Bypass Listener Effe
Bypass Reverb Zone
Play On Awake
Loop
Priority
128
High
Low
Volume
1
Pitch
1
Stereo Pan
0
Left
Right
Spatial Blend
0
2D
3D
Reverb Zone Mix
1
3D Sound Settings
Add Component
ovement...
SampleSc...

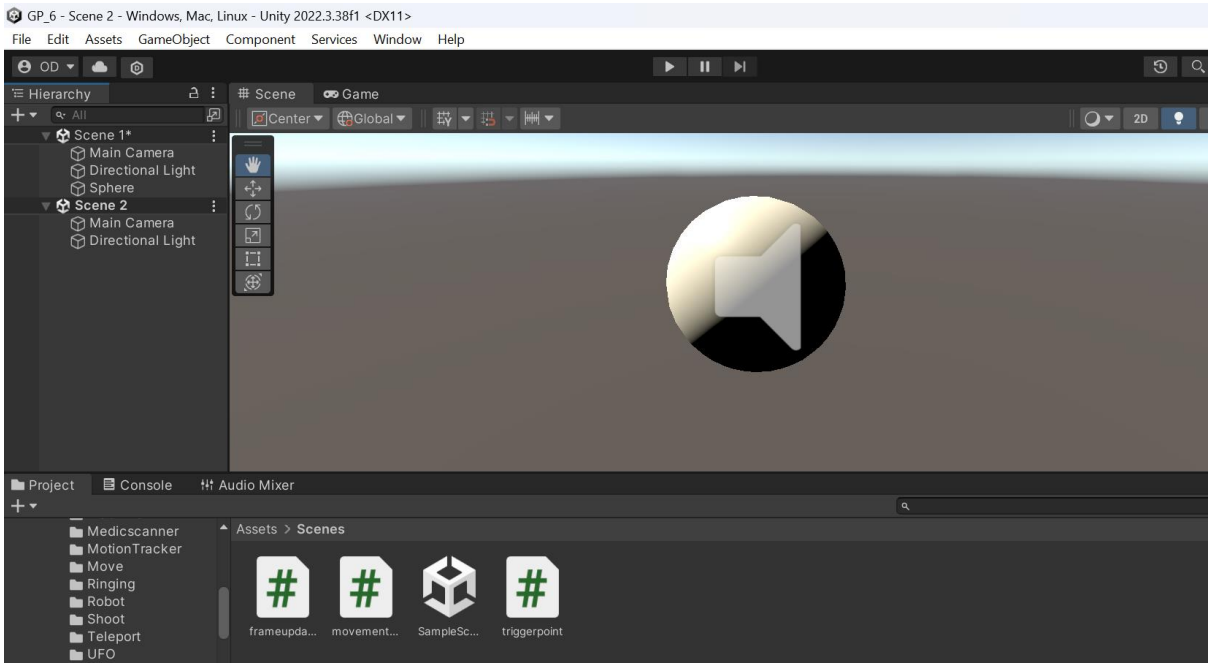
Step 12: Adding the following code as load level script for the trigger point.

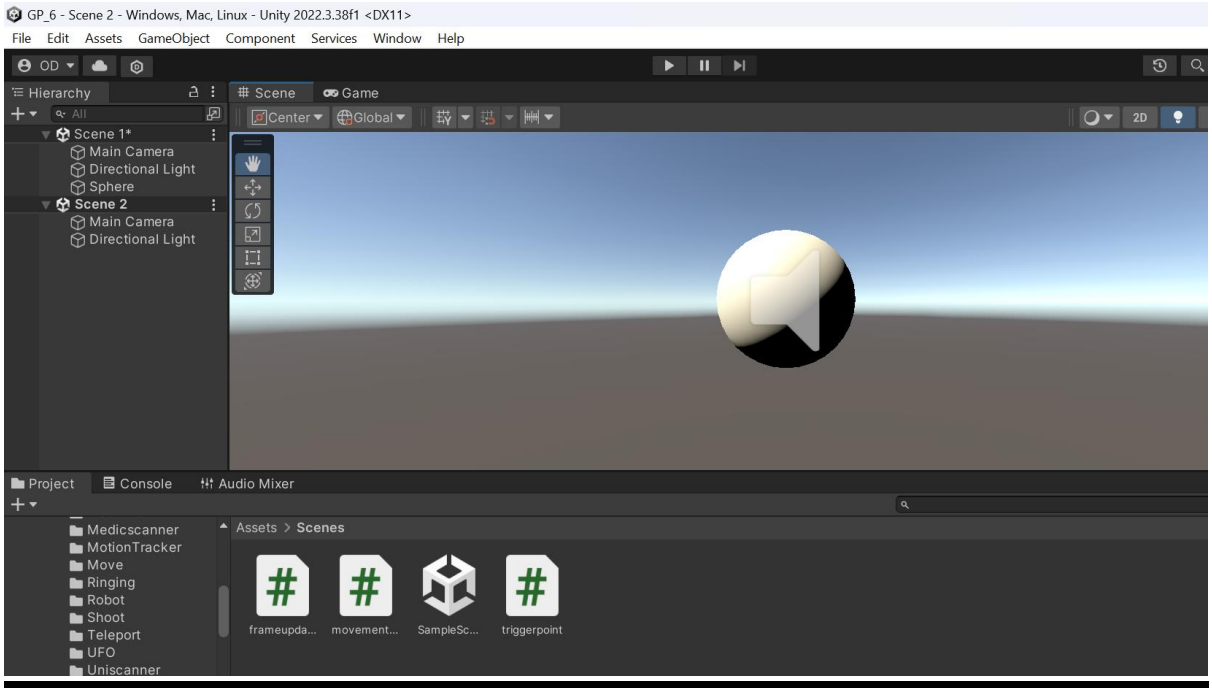
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class LoadLevel : MonoBehaviour
{
    public string level_name;

    private void OnTriggerEnter2D(Collider2D collision)
    {
        GameObject collisionGameObject = collision.gameObject;

        if (collisionGameObject.name == "Player")
        {
            SceneManager.LoadScene(level_name);
        }
    }
}
```


File Edit Selection View Go Run
GP_6
EXPLORER
GP_6
.vscode
Assets
Audio
Scenes
frameupdate.cs
movementsound.cs
triggerpoint.cs
Universe Sounds Fre...
Logs
Packages
UserSettings
Assembly-CSharp.csp...
GP_6.sln
triggerpoint.cs
movementsound.cs
Assets > Scenes > triggerpoint.cs
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
public class LoadLevel : MonoBehaviour
{
public string level_name;
private void OnTriggerEnter2D(Collider2D collision)
{
GameObject collisionGameObject = collision.gameObject;
if (collisionGameObject.name == "Player")
{
SceneManager.LoadScene(level_name);
}
}
}

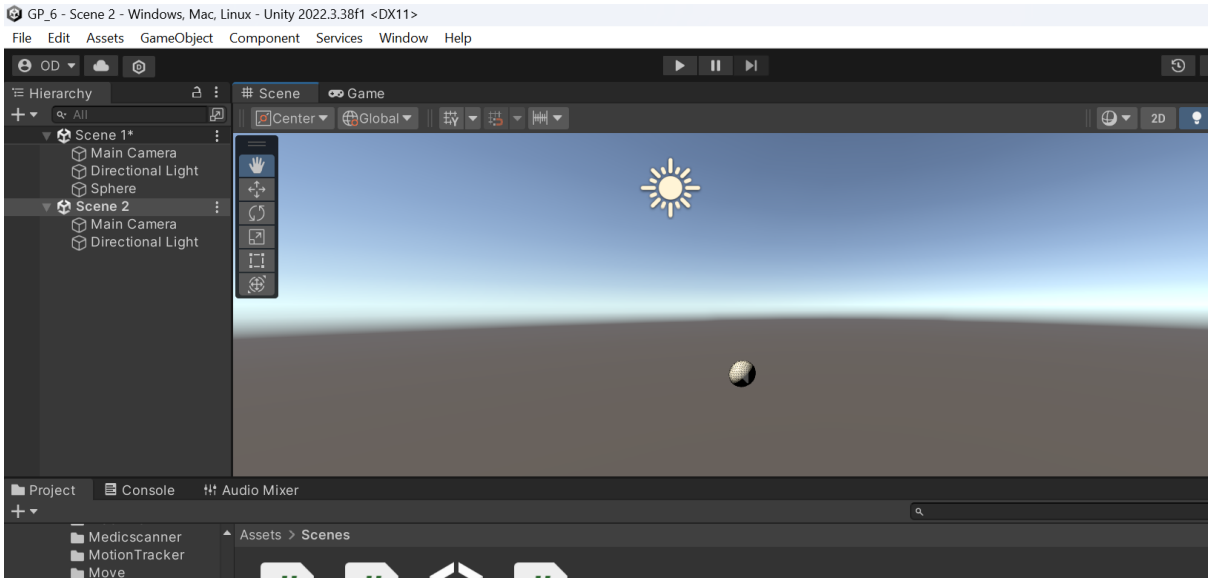
LIGHT EFFECTS PART

Step 13: Creating a similar scene to scene 1 for scene 2, so

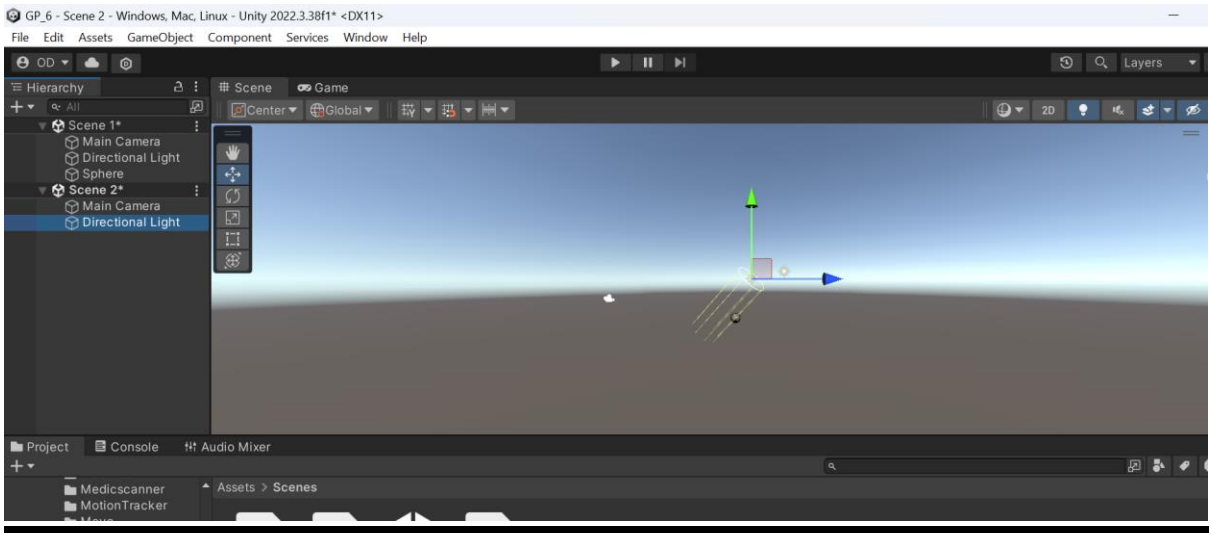




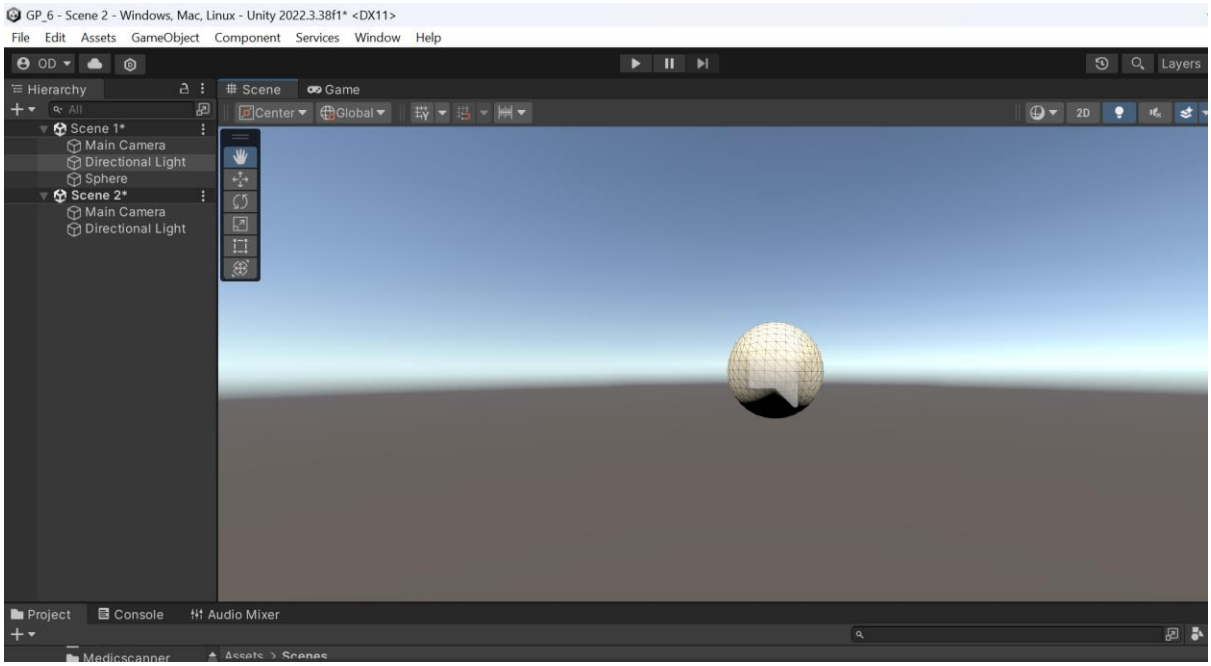
Already in the scene, we can see and tell that there exists a light namely "Directional light"



Step 14: After changing the direction of the directional light we can say that, we get

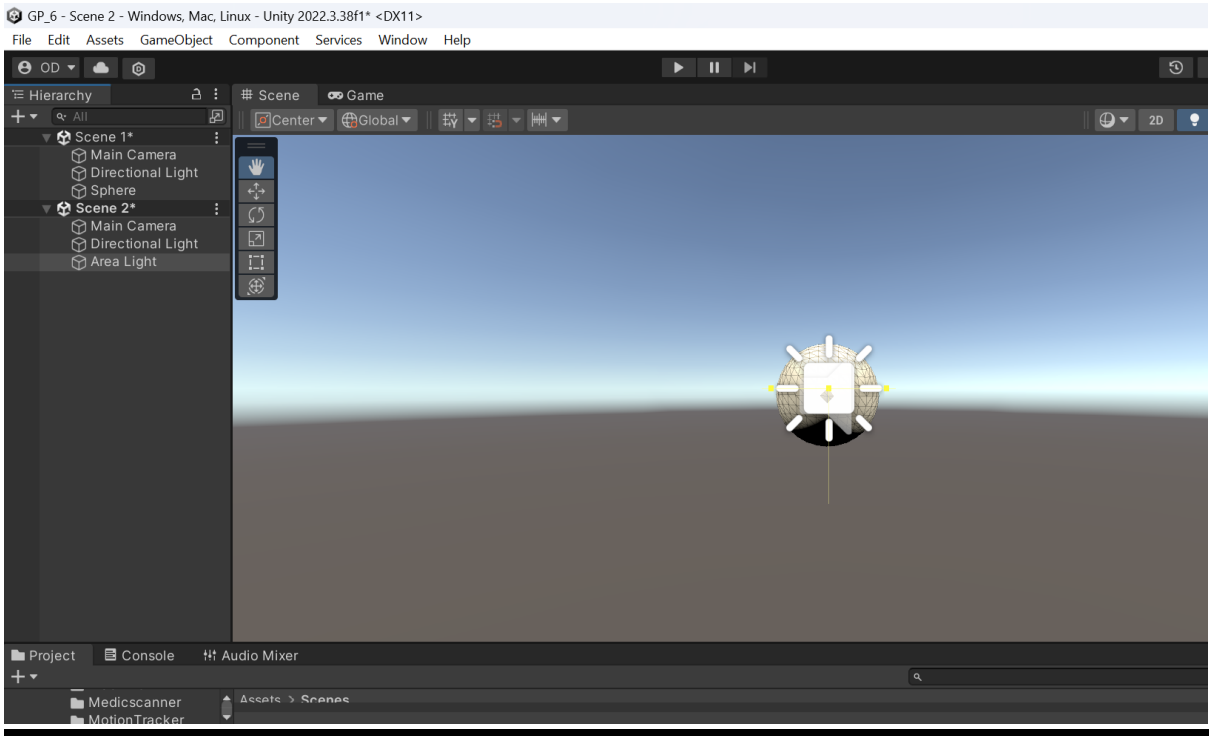


After modifying the direction of directional light

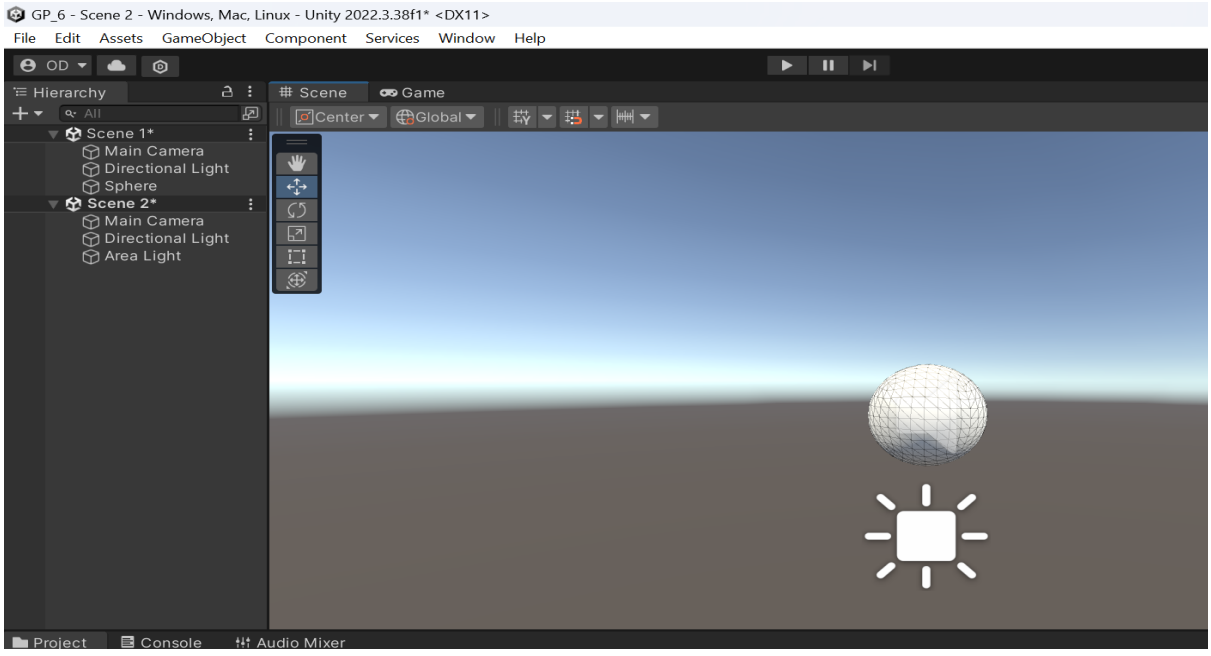


Look in the previous images to see the difference in the part of the sphere where the directional light falls.

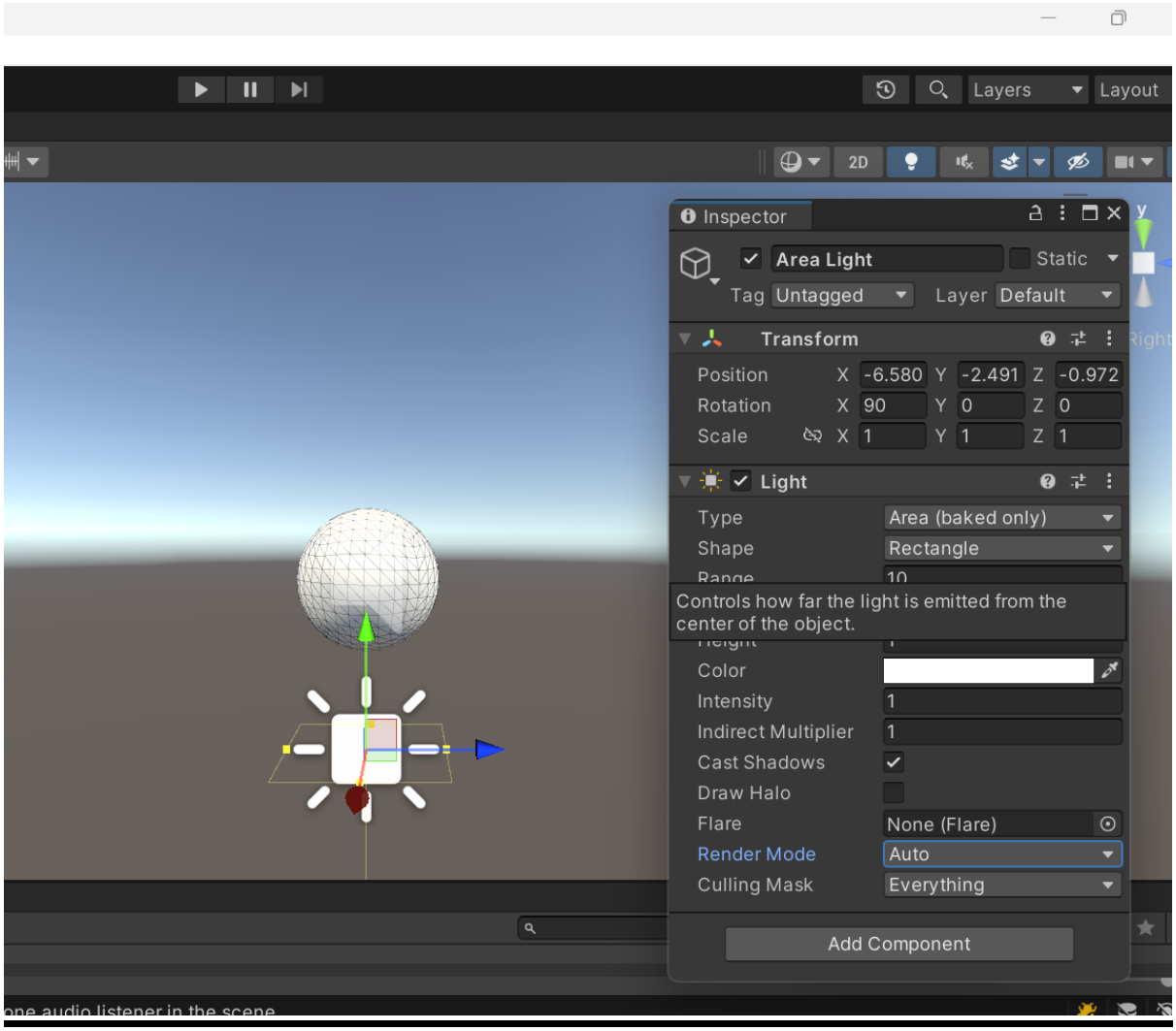
Step 15: Adding Area Light in the scene 2.



Area light when moved to the bottom side of the sphere enlightens the bottom part of sphere which was dark getting no directional light exposure.

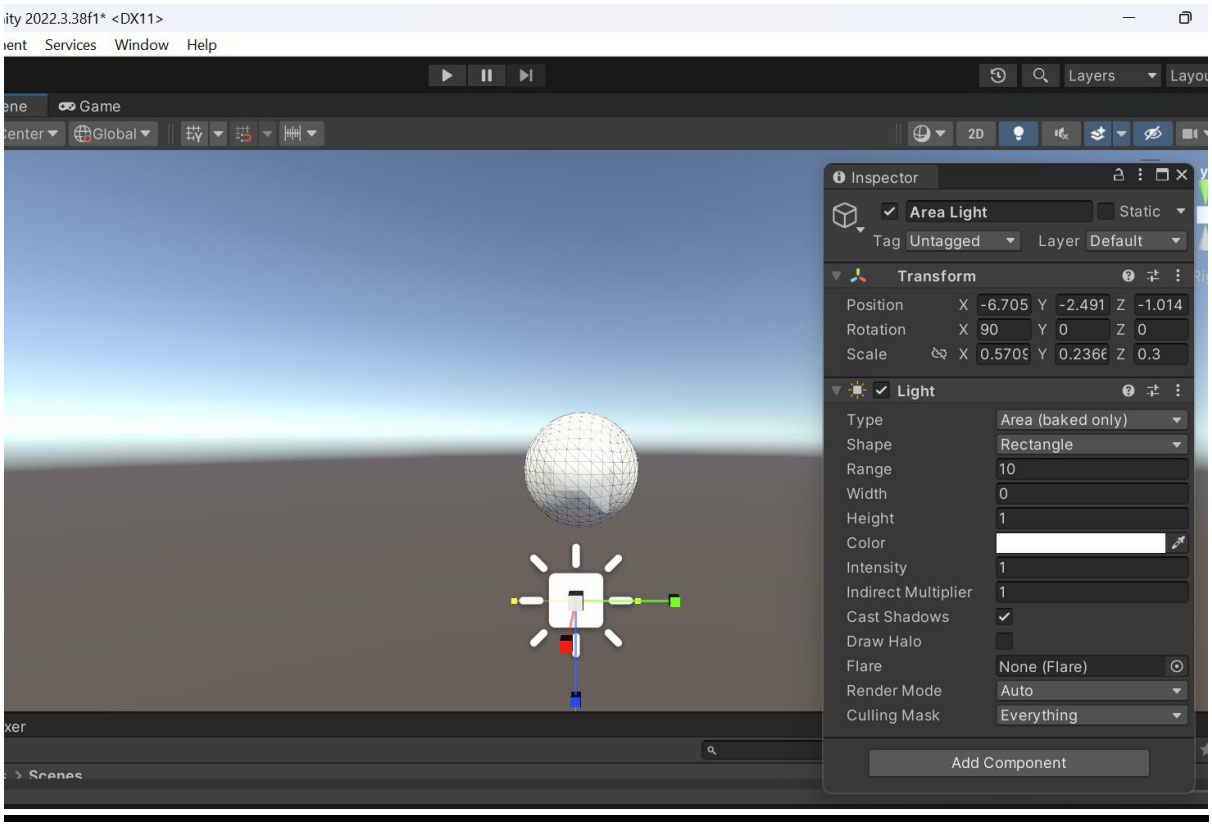


Step 16: Area light selected as (baked mode) type because it will be lighting under that condition only.

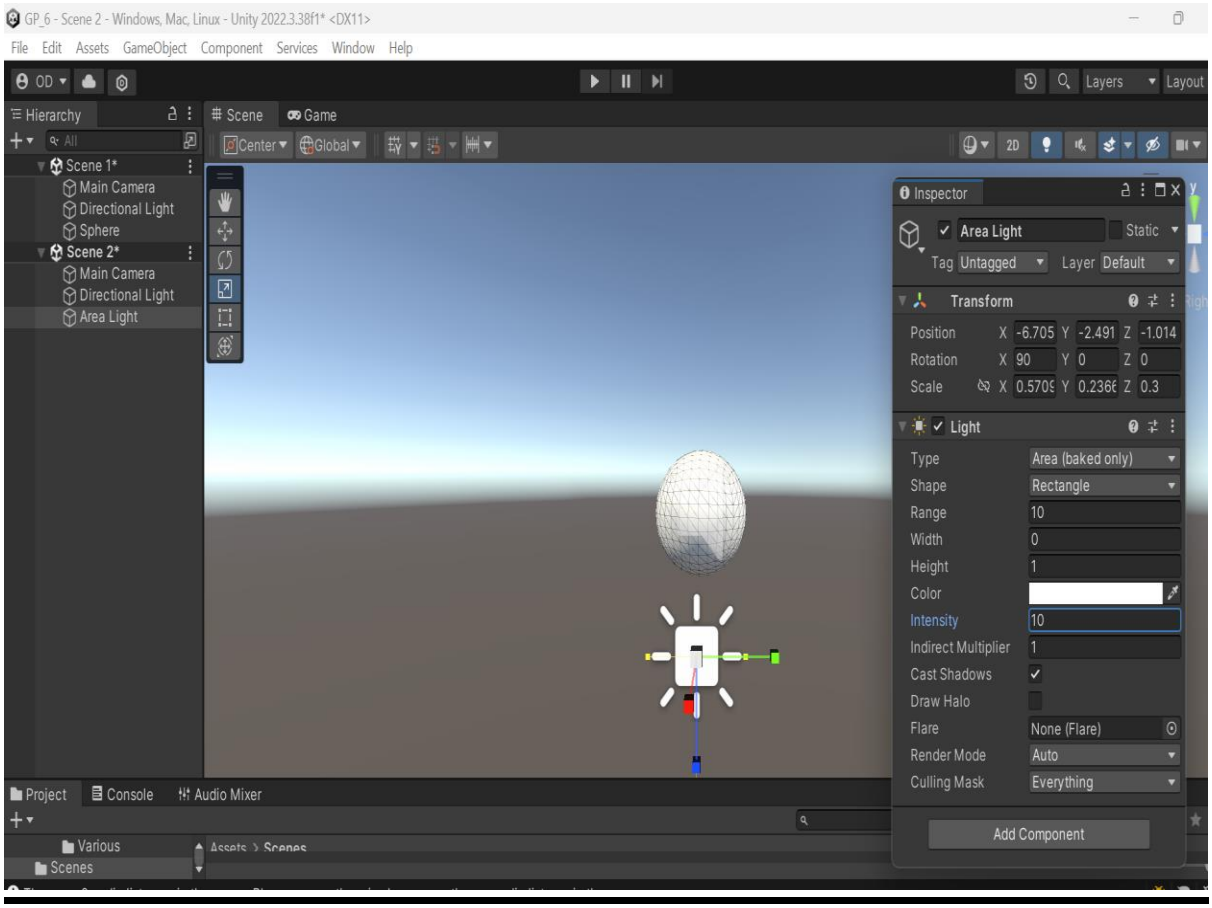


Step 17: Adjusting the size under the values of X,Y,Z under scale.(Difference in values noted from the scale values)

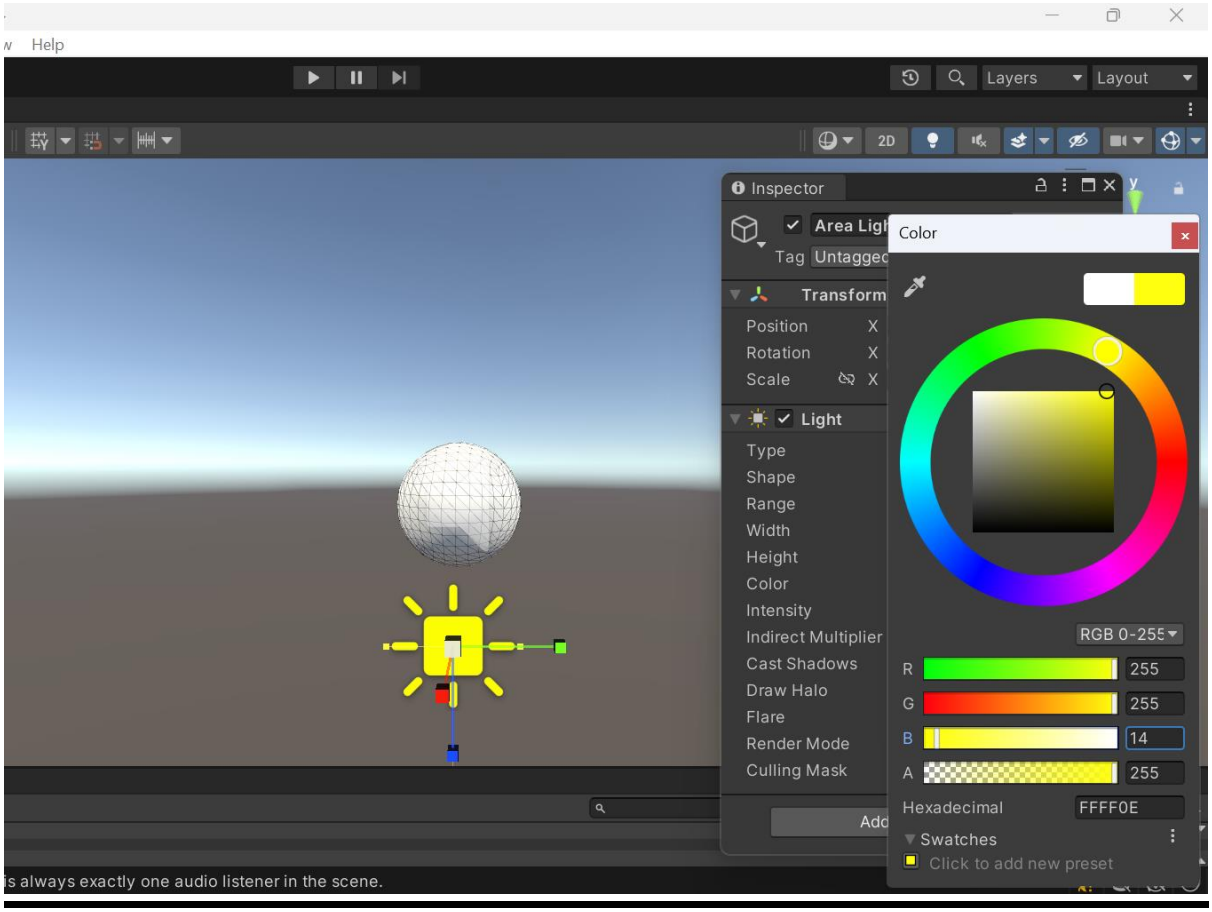
- ➔ Previous values were all equal to 1 for X,Y,Z under the scale boxes.
- ➔ Current values are assigned randomly between 0 and 1.



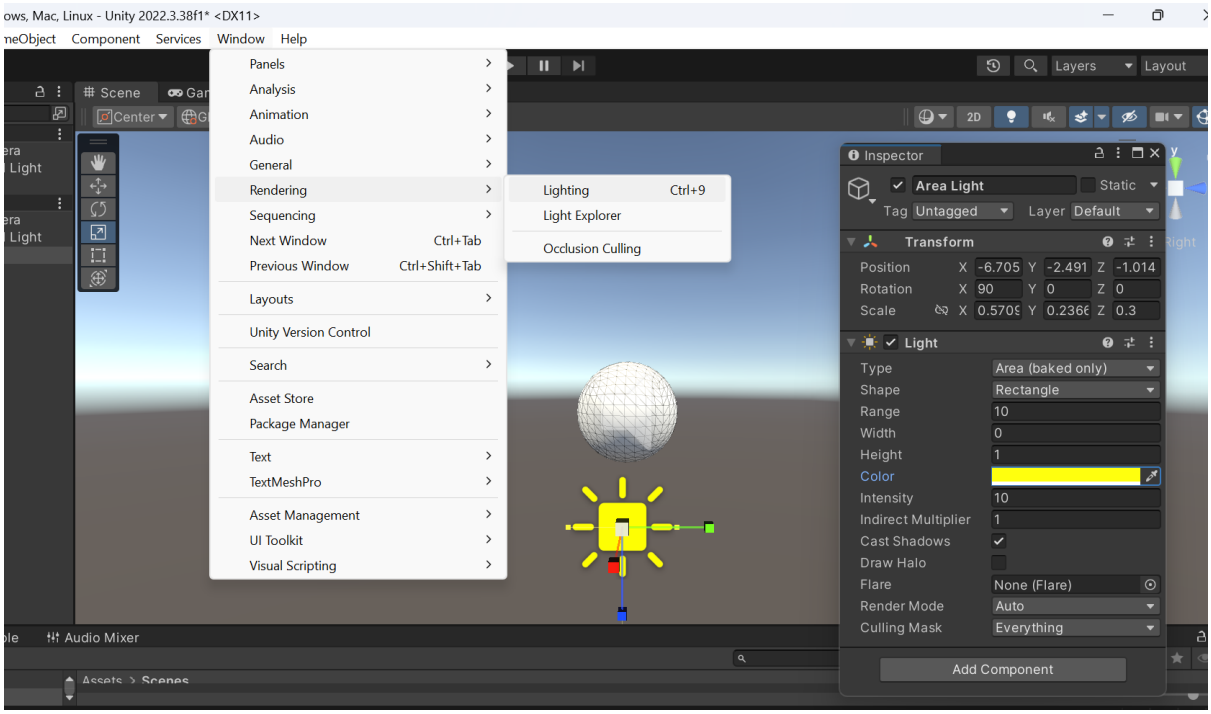
Step 18: Increase the intensity of Area Light from 1 to 10 to show the difference.

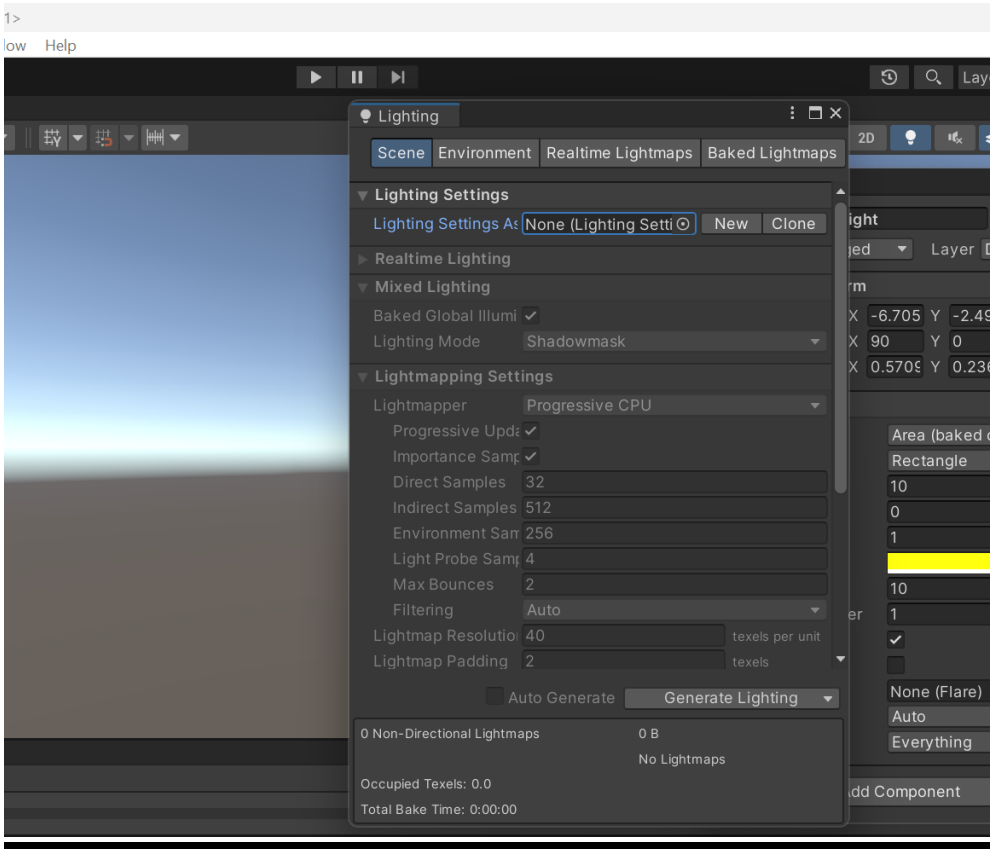


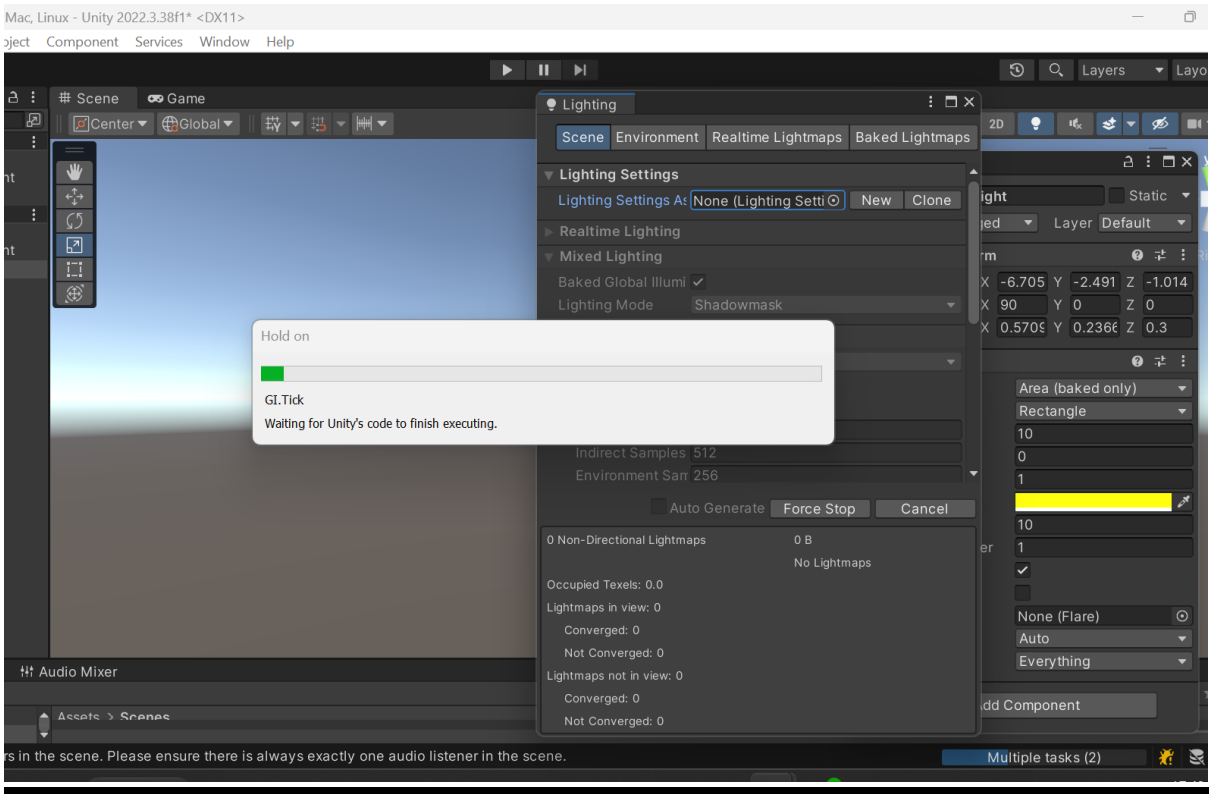
Step 19: Changing the colour of the area light to any favourable colour.



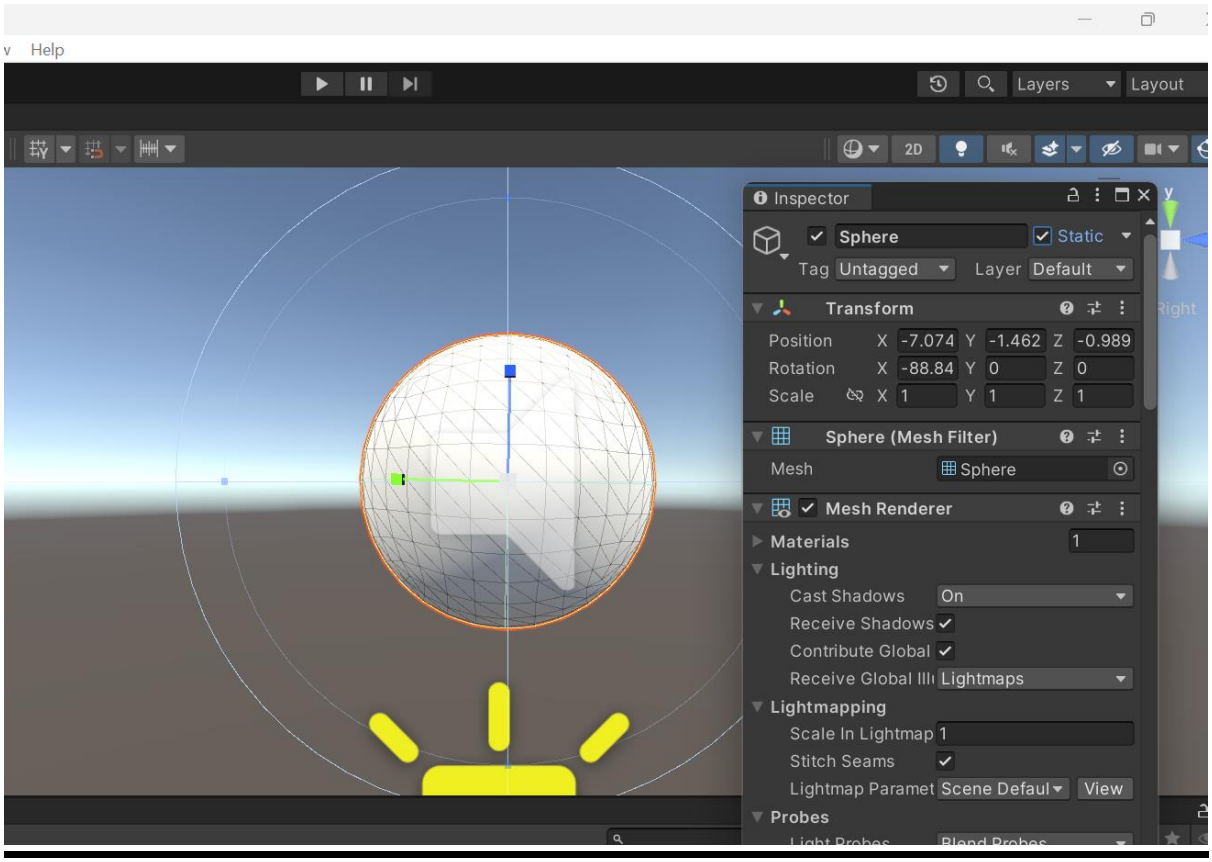
Step 20: Settting scene for baked lighting

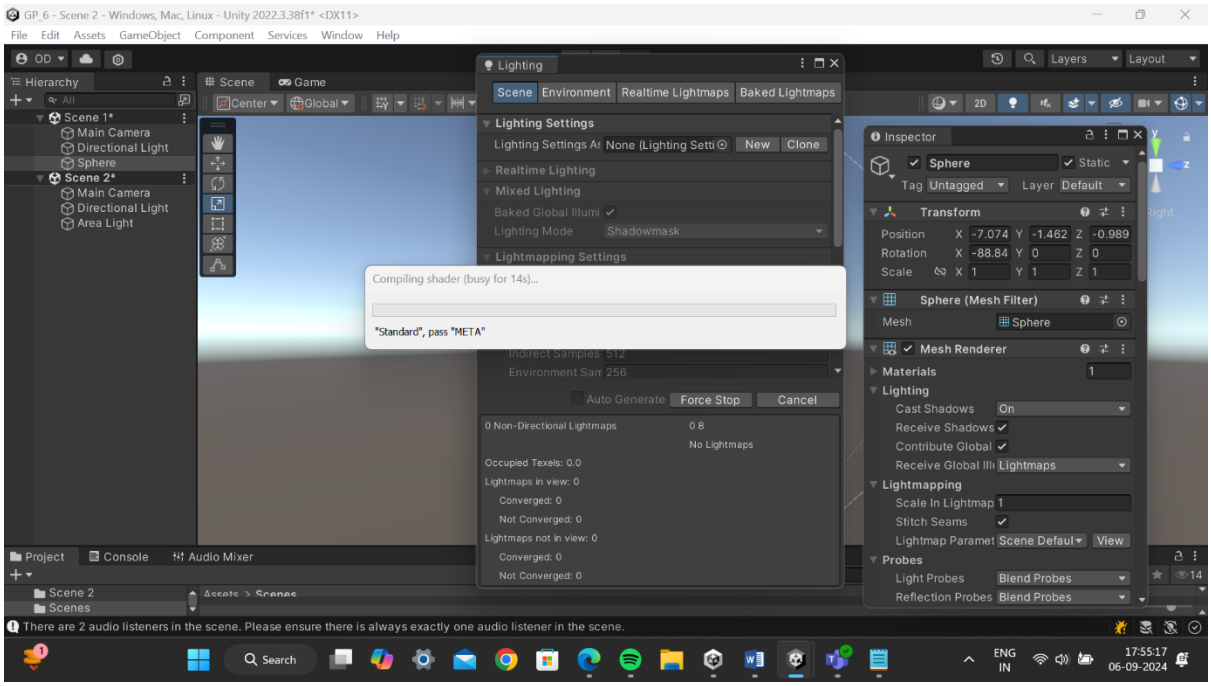




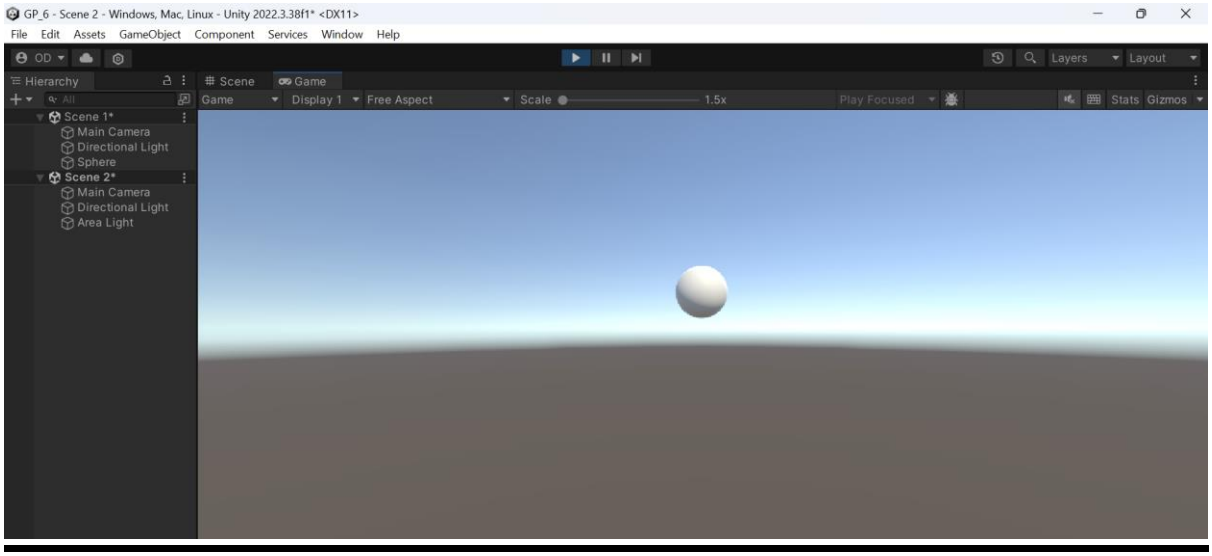


Step 21: Bake the light(selecting object sphere as static)

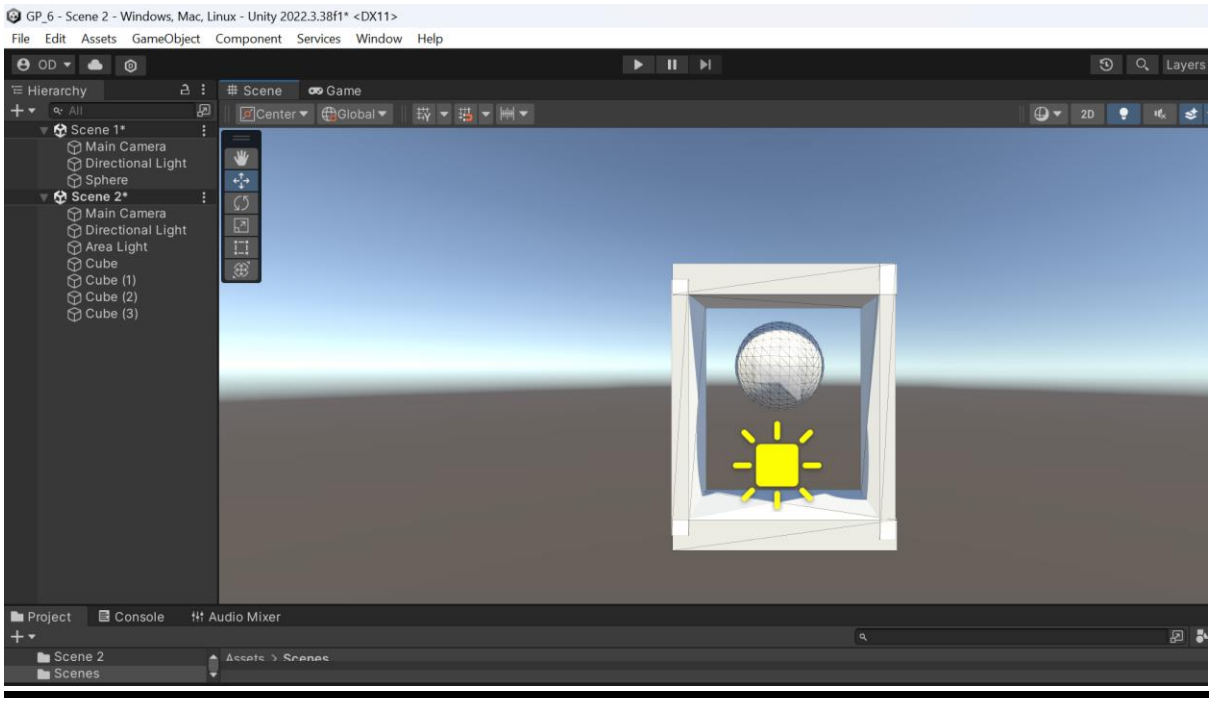




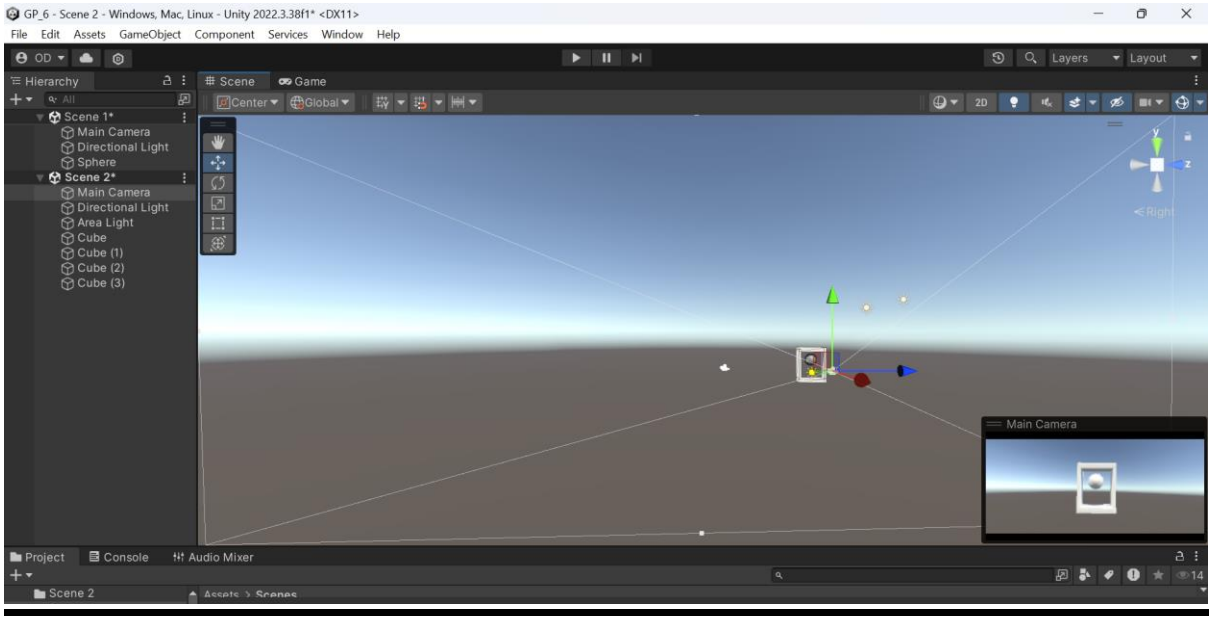
Step 22: Visualizing the effect (before adding more objects)

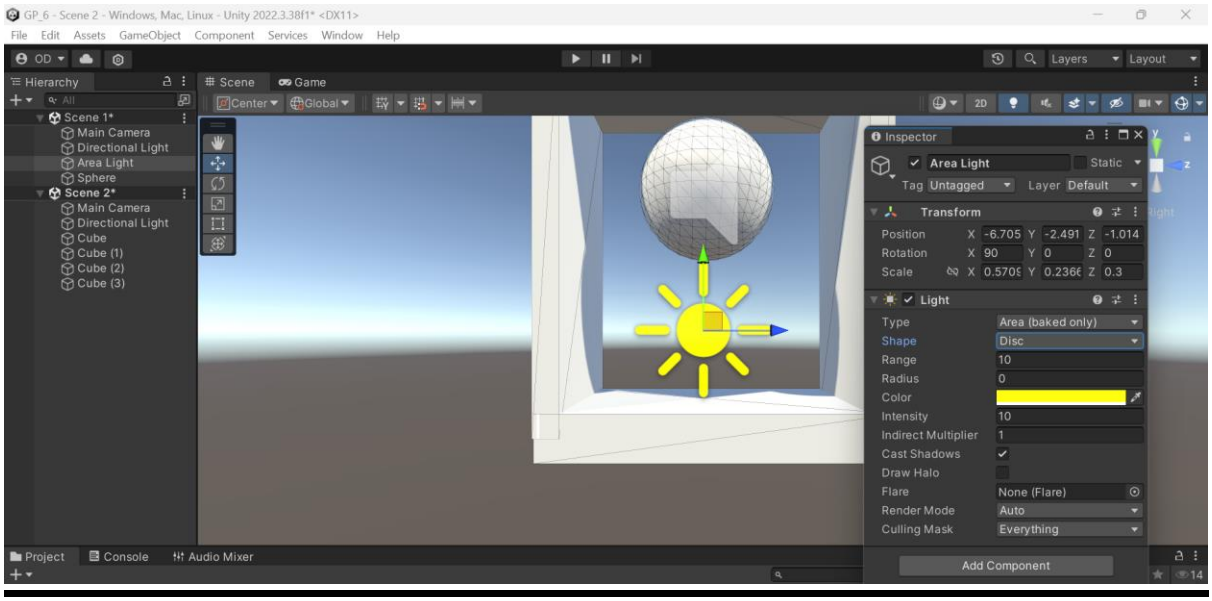


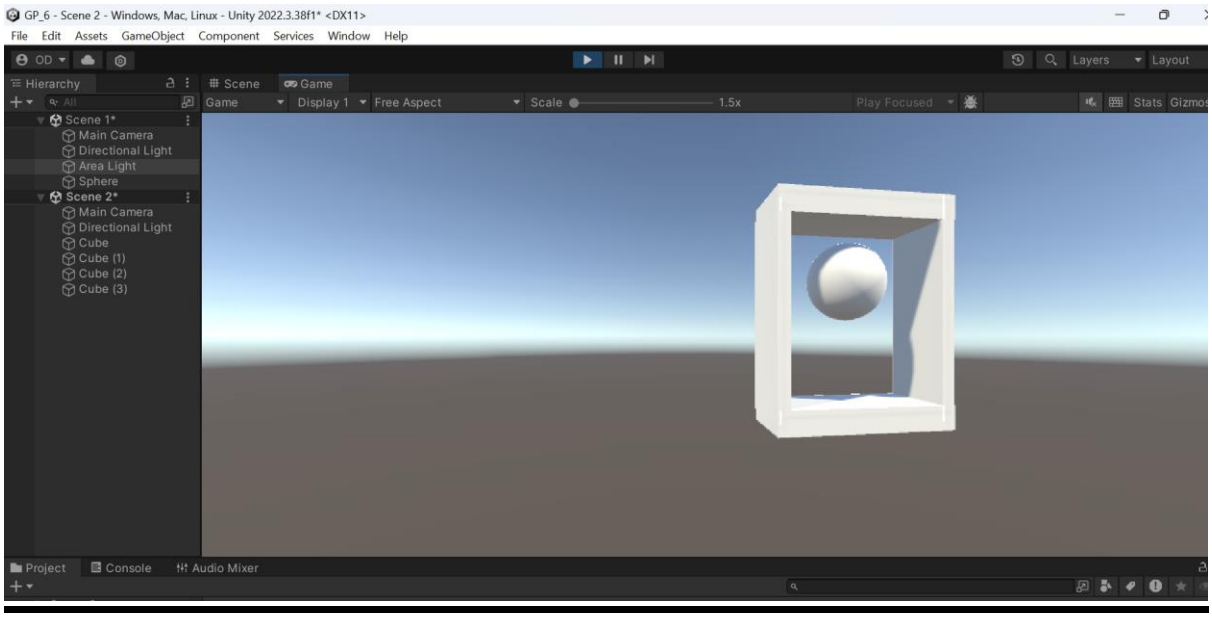
Step 23: Manipulating the indirect multiplier and then adding different objects to test the scenario.(after adding more objects)



Setting the main camera accordingly:




GP_6 - Scene 2 - Windows, Mac, Linux - Unity 2022.3.38f1* <DX11>
File Edit Assets GameObject Component Services Window Help
Hierarchy
Scene 1*
Main Camera
Directional Light
Area Light
Sphere
Scene 2*
Main Camera
Directional Light
Cube
Cube (1)
Cube (2)
Cube (3)
Scene
Game
Inspector
Area Light
Static
Tag Untagged
Layer Default
Transform
Position X -6.705 Y -2.491 Z -1.014
Rotation X 90 Y 0 Z 0
Scale X 0.5709 Y 0.2366 Z 0.3
Light
Type Area (baked only)
Shape Disc
Range 10
Radius 0
Color
Intensity 10
Indirect Multiplier 1
Cast Shadows
Draw Halo
Flare None (Flare)
Render Mode Auto
Culling Mask Everything
Add Component
Project
Console
Audio Mixer


GP_6 - Scene 2 - Windows, Mac, Linux - Unity 2022.3.38f1* <DX11>
File Edit Assets GameObject Component Services Window Help
Hierarchy
Scene 1*
Main Camera
Directional Light
Area Light
Sphere
Scene 2*
Main Camera
Directional Light
Cube
Cube (1)
Cube (2)
Cube (3)
Scene
Game
Game
Display 1
Free Aspect
Scale
1.5x
Play Focused
Stats
Project
Console
Audio Mixer