

Students Name: Aniket Yadav

Roll No: 5734

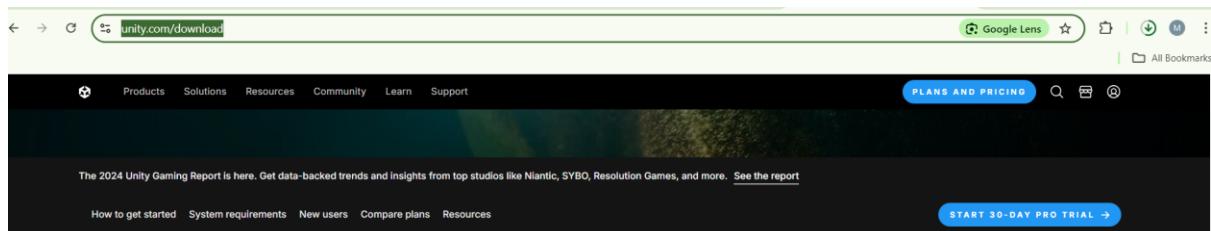
Subject: Augmented Reality & Virtual Reality

Class: MCS Part 2

Academic Year: 2024-2025

Practical 1: Install unity.

Step 1: Go to the official Unity website <https://unity.com/download>. Click download Unity Hub.



Create with Unity in three steps

1. Download the Unity Hub

Follow the instructions onscreen for guidance through the installation process and setup.

[Download for Windows](#)
[Download for Mac](#)
[Instructions for Linux](#)

2. Choose your Unity version

Install the latest version of Unity, an older release, or a beta featuring the latest in-development features.

[Visit the download archive](#)

3. Start your project

Begin creating from scratch, or pick a template to get your first project up and running quickly. Access tutorial videos designed to support creators, from beginners to experts.

[Access our Pro Onboarding Guide](#)

Step 2: Run the Unity Hub installer that you just downloaded.

Step 3: Once installed, open Unity Hub. You may need to sign in or create a Unity account if you don't have one.

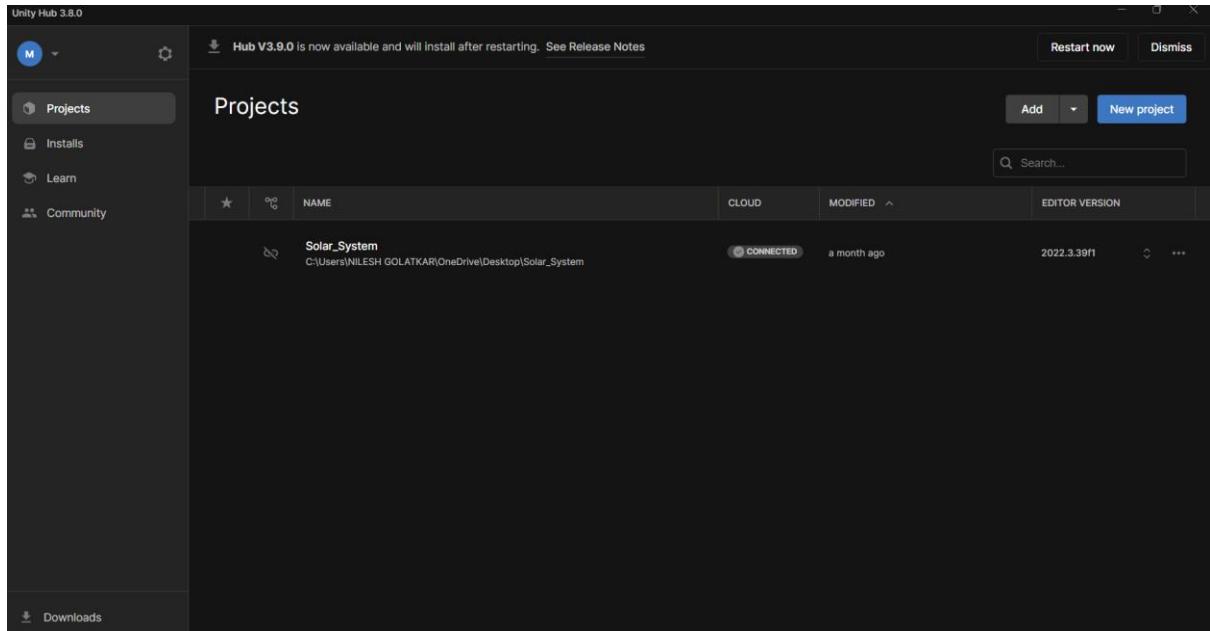
Create a Unity ID

If you already have a Unity ID, please [sign in](#).

Email	Password
Username	Full Name
<input type="checkbox"/> I have read and agree to the Unity Terms of Service (required). <input type="checkbox"/> I acknowledge the Unity Privacy Policy [Republic of Korea Residents agree to the Unity Collection and Use of Personal Information (required).] <input type="checkbox"/> I agree to have Marketing Activities directed to me by and receive marketing and promotional information from Unity, including via email	
<input type="checkbox"/> I'm not a robot  <small>reCAPTCHA Privacy - Terms</small>	
By clicking "Accept Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.	
Cookie Settings Reject All Accept All Cookies	

Step 4: In Unity Hub, go to the Installs tab. Click Add to choose the version of Unity you want to install. Select the Unity version and components (such as Android Build Support, iOS Build Support) that you need. Click Install and wait for the installation to complete.

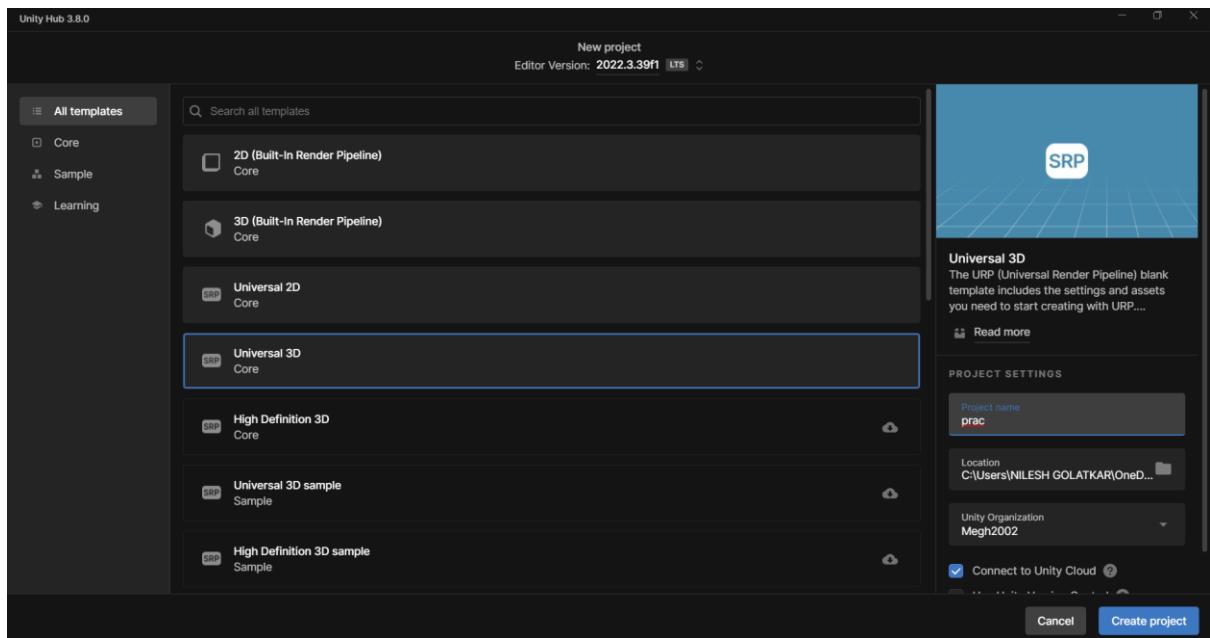
Step 5: Go to the Projects tab in Unity Hub. Click New Project to create a new Unity project. Choose a template (e.g., 2D, 3D) and set the project location, then click Create.



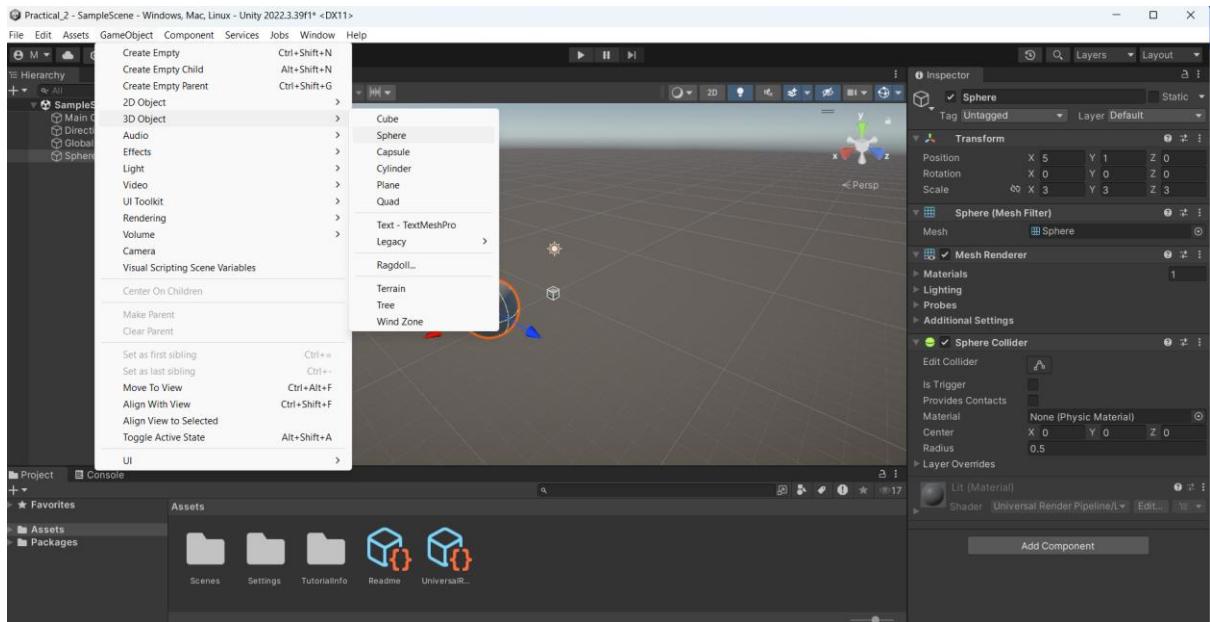
Step 6: After the project is created, Unity will automatically open, and you can start developing your project.

Practical 2: Changing colour of object on button click in unity.

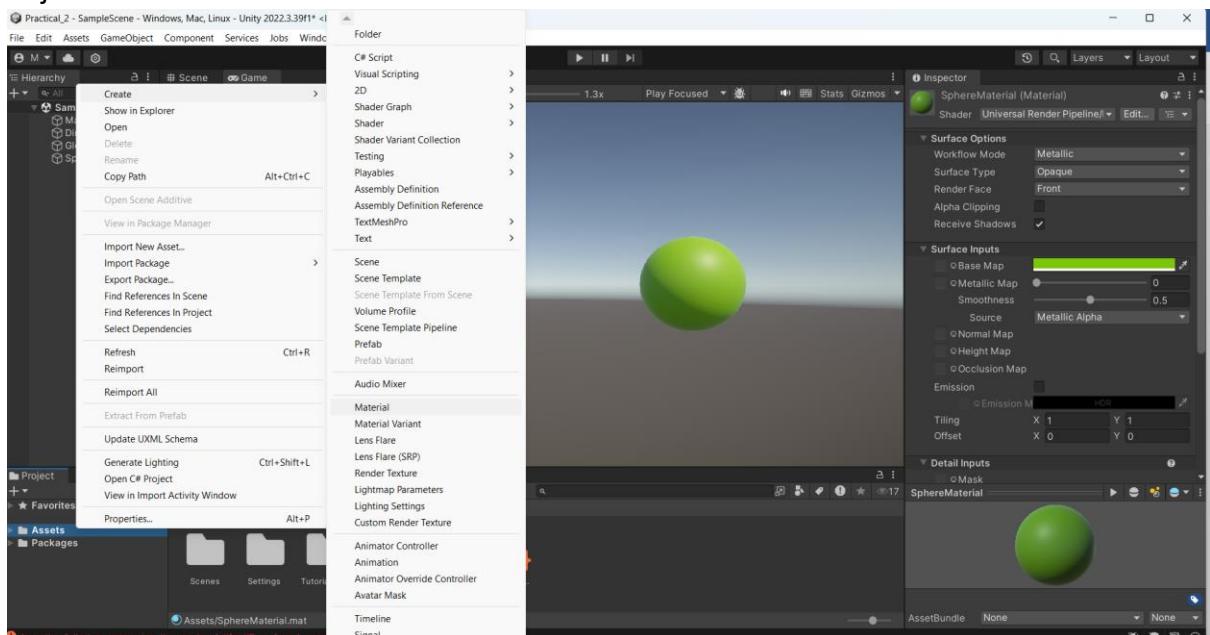
Step 1: Open unity, create new project, select 3d universal templates, give project name and click create project.



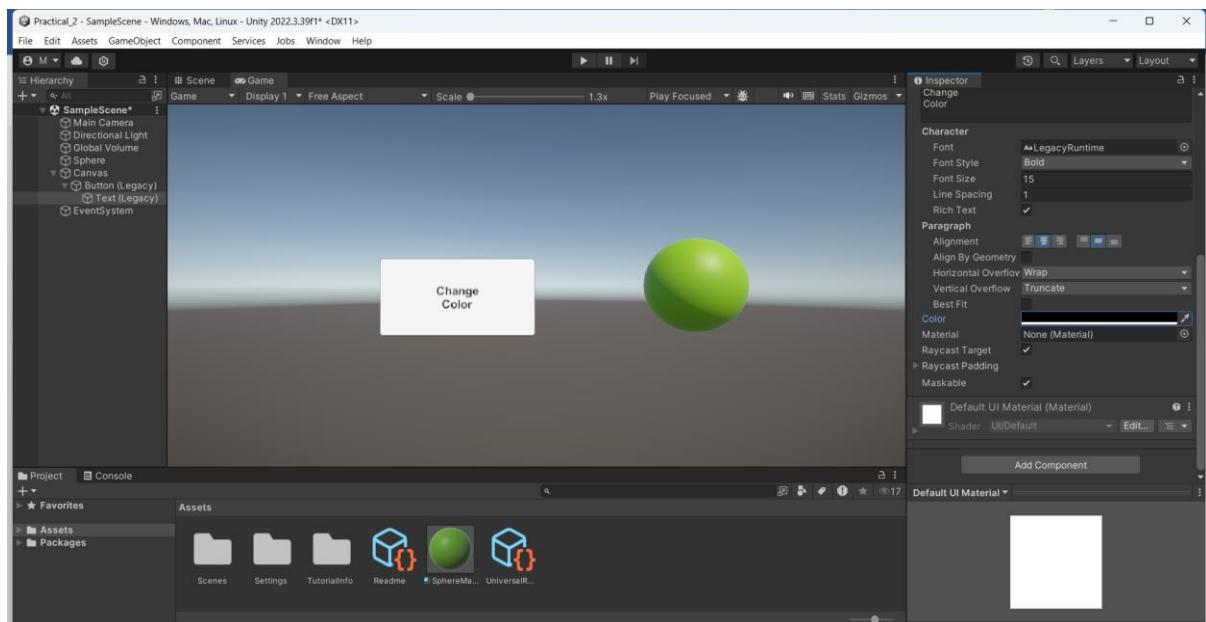
Step 2: Click on GameObject and select 3d object, choose any object. Change position and scale by changing transform.



Step 3: Right click on assets -> create -> material -> change name of material "spherematerial" -> change the color of object in inspector panel and drag the material on object.



Step 4: Click on GameObject -> UI -> button. Import both the packages. Change the position and size of button in inspector panel. Under the button there is 'text'. In that you can change text of button, color, size of text.



Step 5: Right click on assets -> create -> c# script -> change the name of script "changecolor". Add following code and drag this script on object. Go to inspector panel of button click + of on click drag the object in none object and change no function into change colors script -> change color(). Drag the sphere object on script of button inspector.

using UnityEngine;

```
public class ChangeColorScript : MonoBehaviour
```

```
{
```

```
// Method to change the color
```

```
public void ChangeColor()
```

```
{
```

```

// Generate a random color

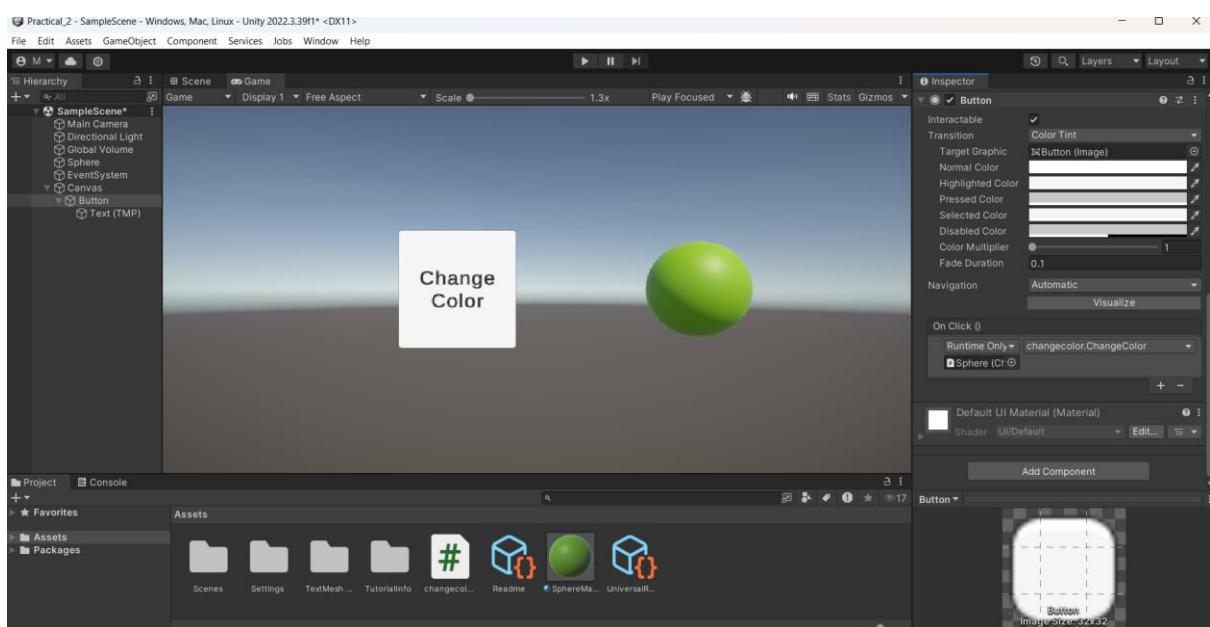
Color randomColor = new Color(Random.value, Random.value, Random.value);

// Apply the color to this object's material

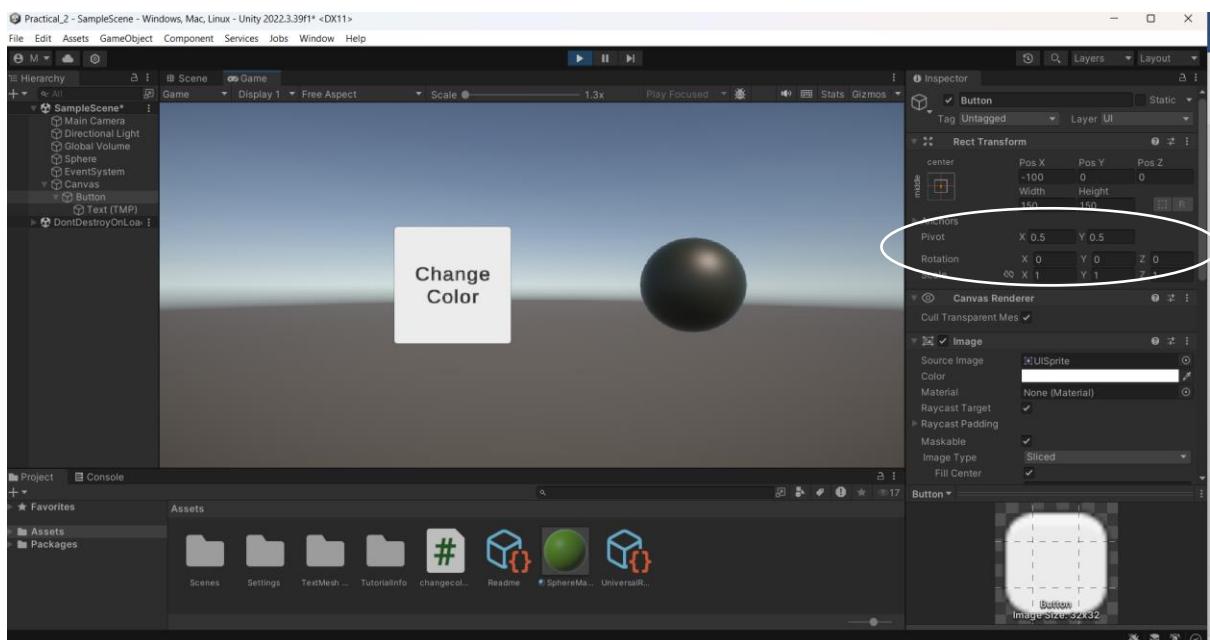
GetComponent<Renderer>().material.color = randomColor;

}
}

```



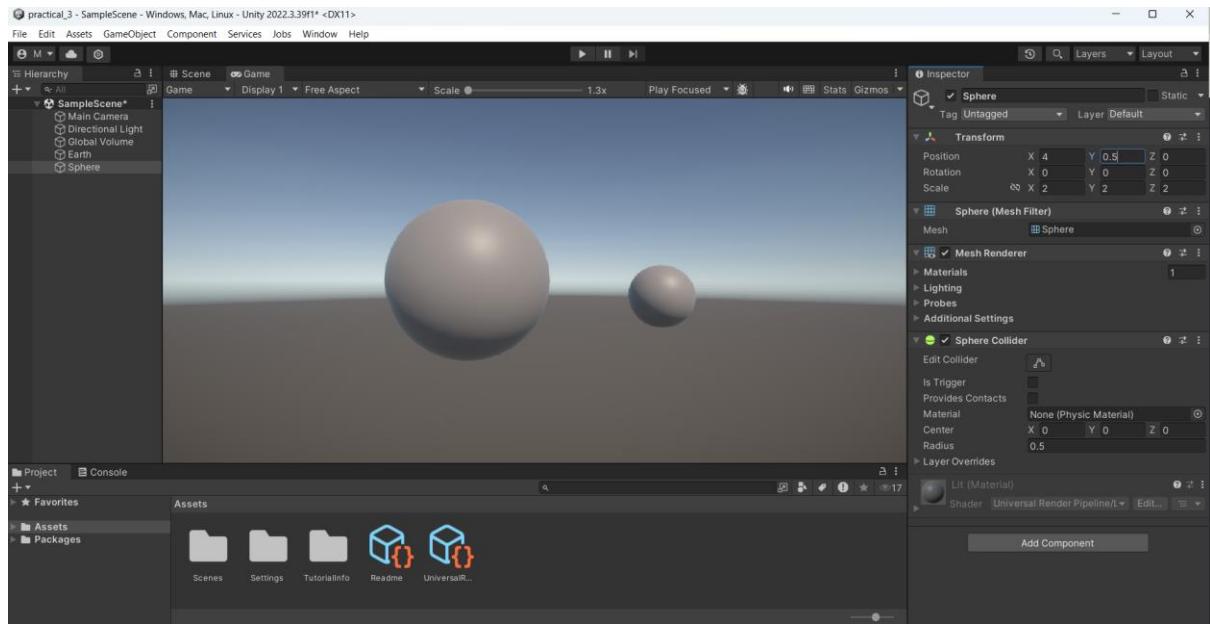
Step 6: Run the project.



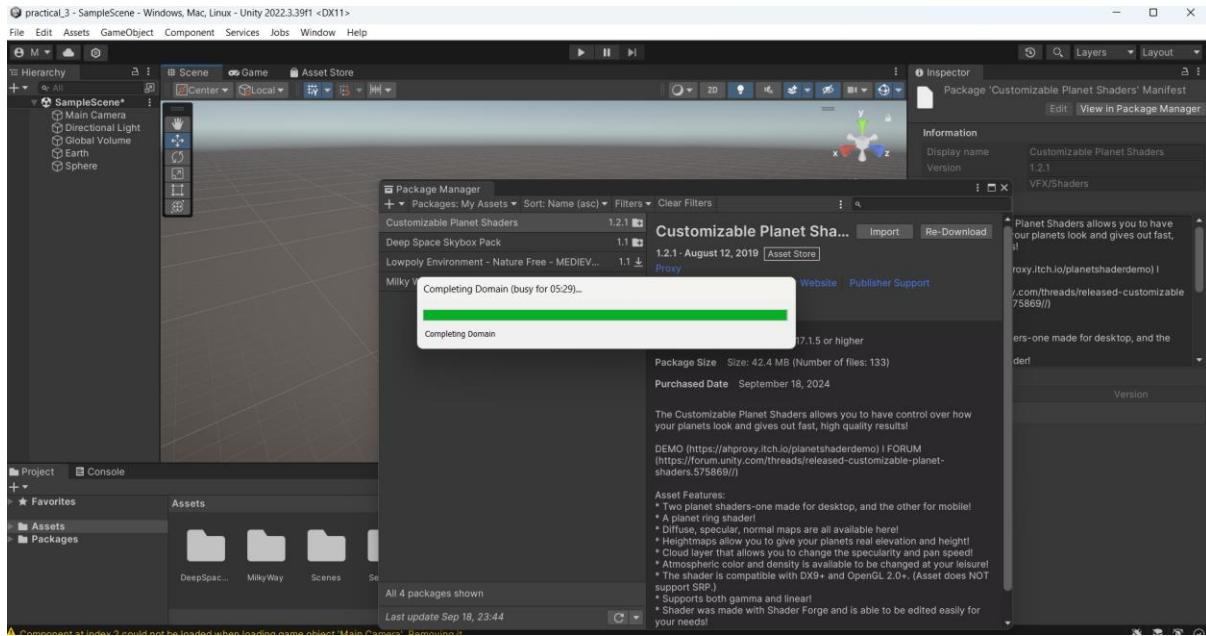
Practical 3: Changing the texture of object.

Step 1: Open unity, create a new project.

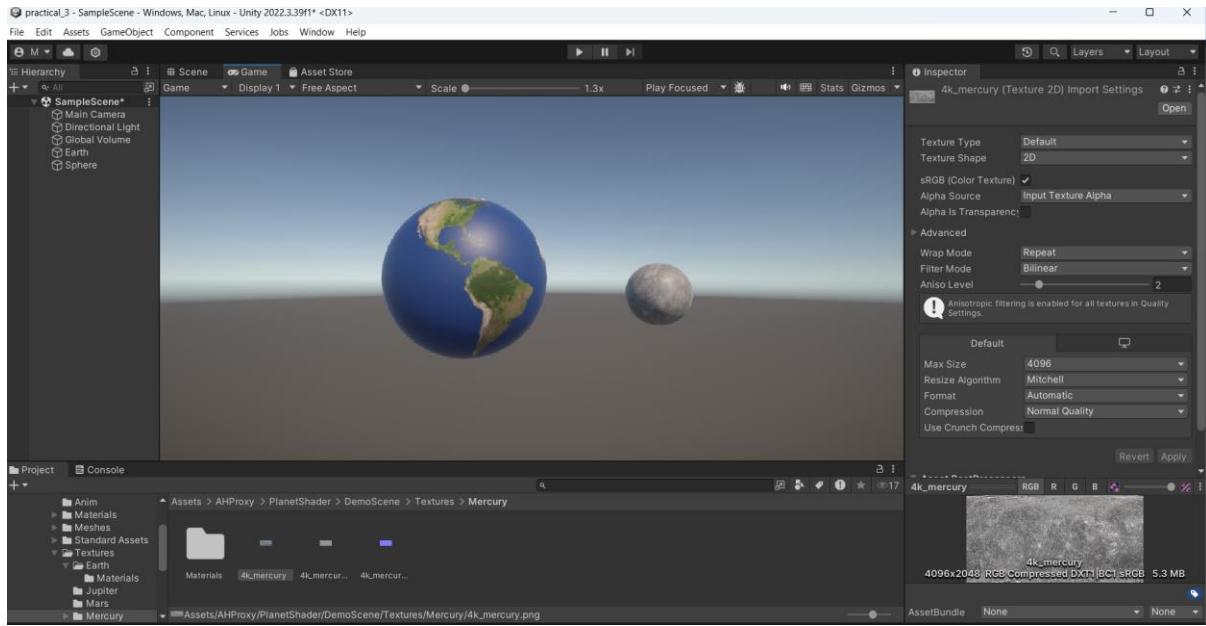
Step 2: Game object -> 3d object -> Sphere -> Change transform and size -> Take another sphere change transform and size.



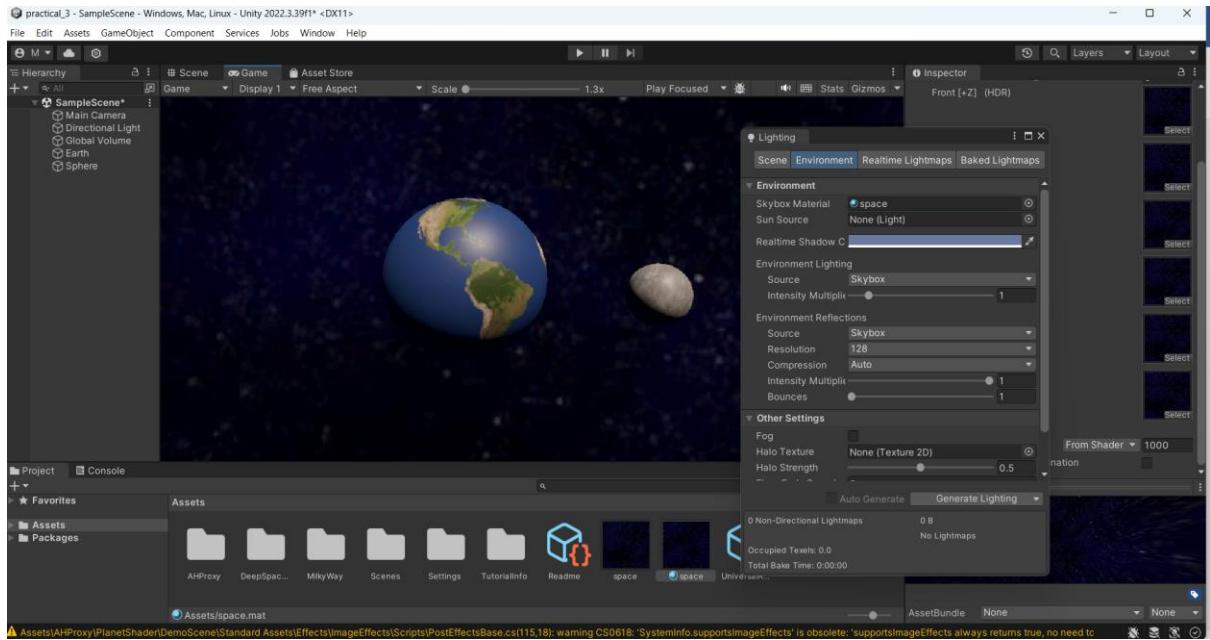
Step 3: Go to unity assets store download packages of textures that you want to apply on object and import it into project.



Step 4: Drag the Earth texture from assets on big sphere as well as moon texture on small sphere.



Step 5: Create material in assets name as space. Change shader as skybox in inspector panel. Window -> Rendering -> Environment -> Drag created space material on skybox material.



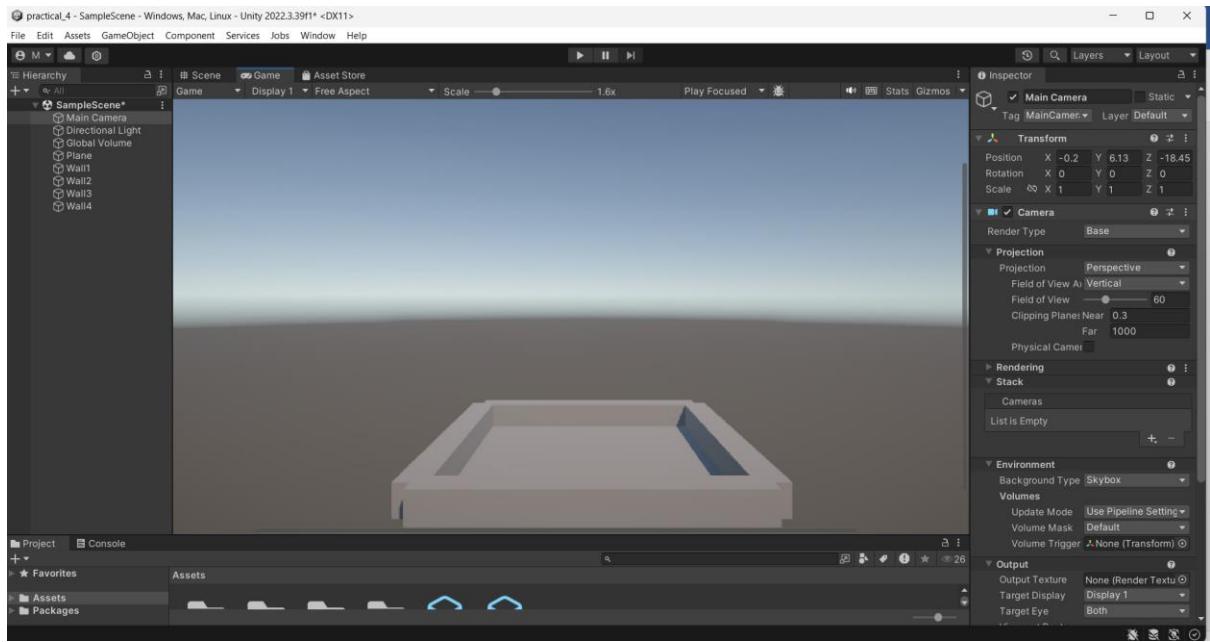
Step 6: Run the project.



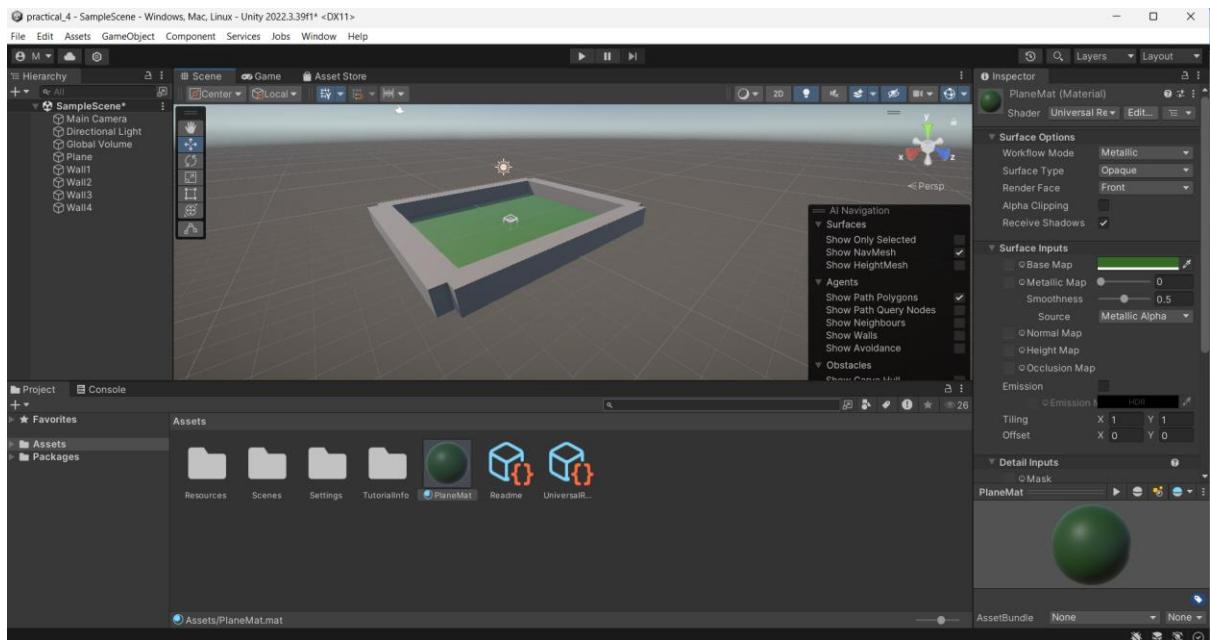
Practical 4: Creating a plane and placing cube and sphere applying transformation.

Step 1: Open unity, create a new project.

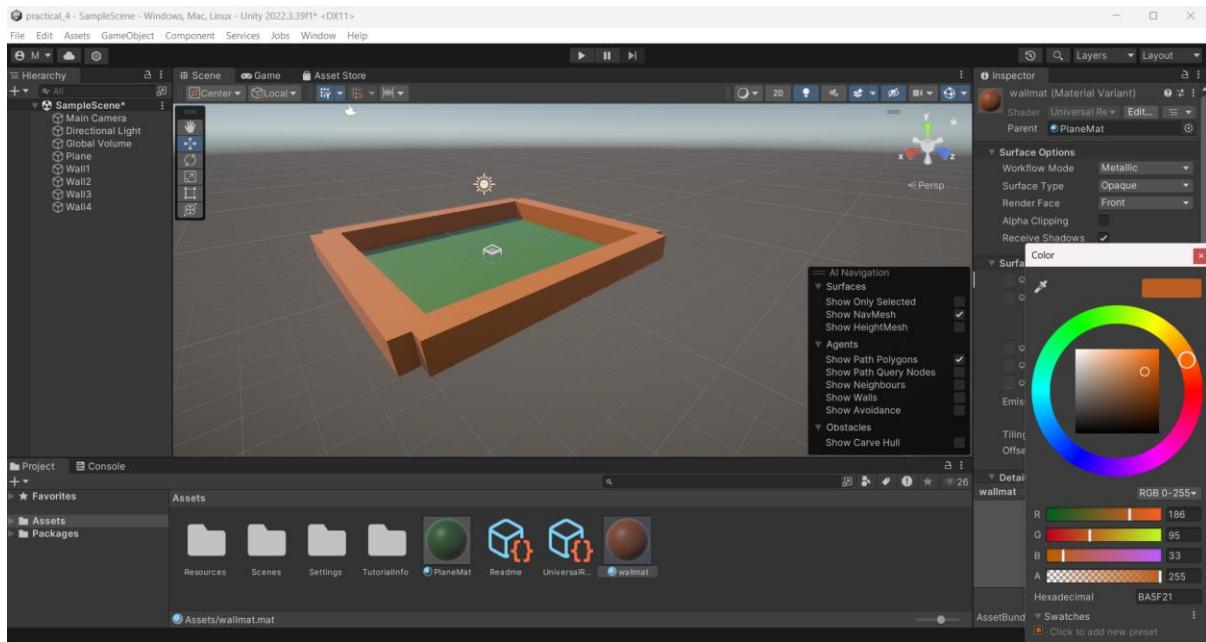
Step 2: Gameobject -> 3d object -> plane and change the transformation -> 3d object -> take 4 cubes and change the transformation.



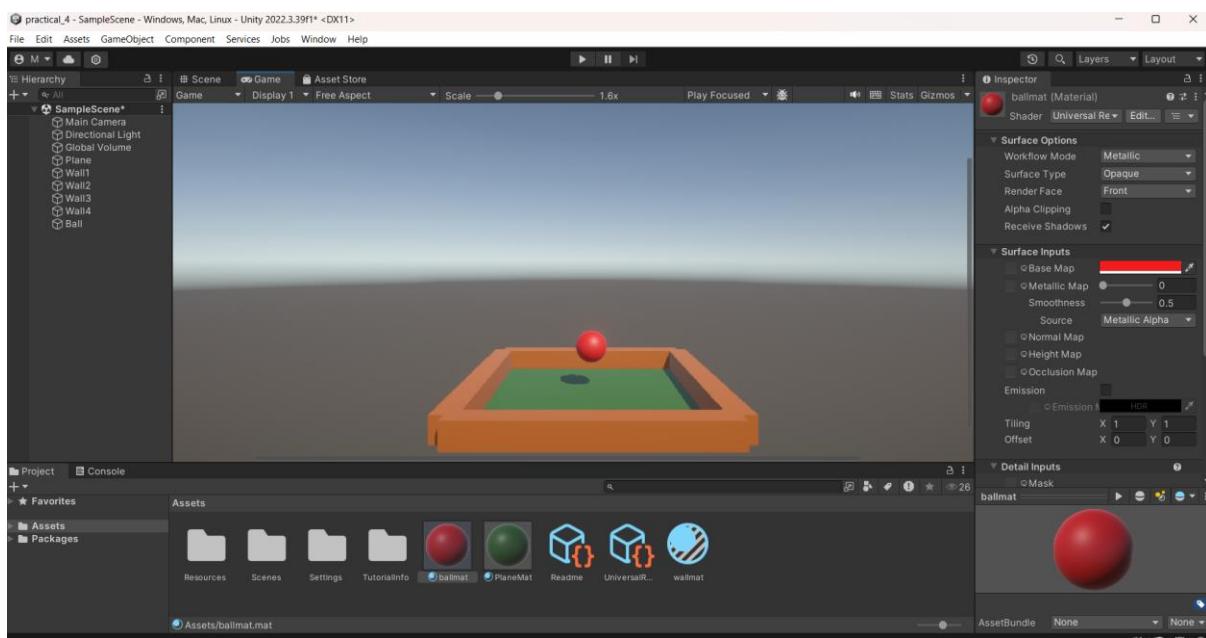
Step 3: Right click on assets -> create -> material -> Rename material as planemat -> Change color of material in inspector panel -> Drag that material on plane.



Step 4: Right click on assets -> create -> material -> Rename material as wallmat -> Change color of material in inspector panel -> Drag that material on all wall.



Step 5: Gameobject -> 3d object -> Sphere -> Change transformation -> Create new ballmat -> Change color.



Step 6: in inspector panel of ball -> add component -> rigidbody -> run. Assets right click -> c# script -> apply script on ball -> change the speed of script in inspector panel.

```
using System.Collections;
```

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

```
using UnityEngine;
```

```
public class ballmovement : MonoBehaviour
```

```
{
    public float Speed;

    void FixedUpdate()
    {

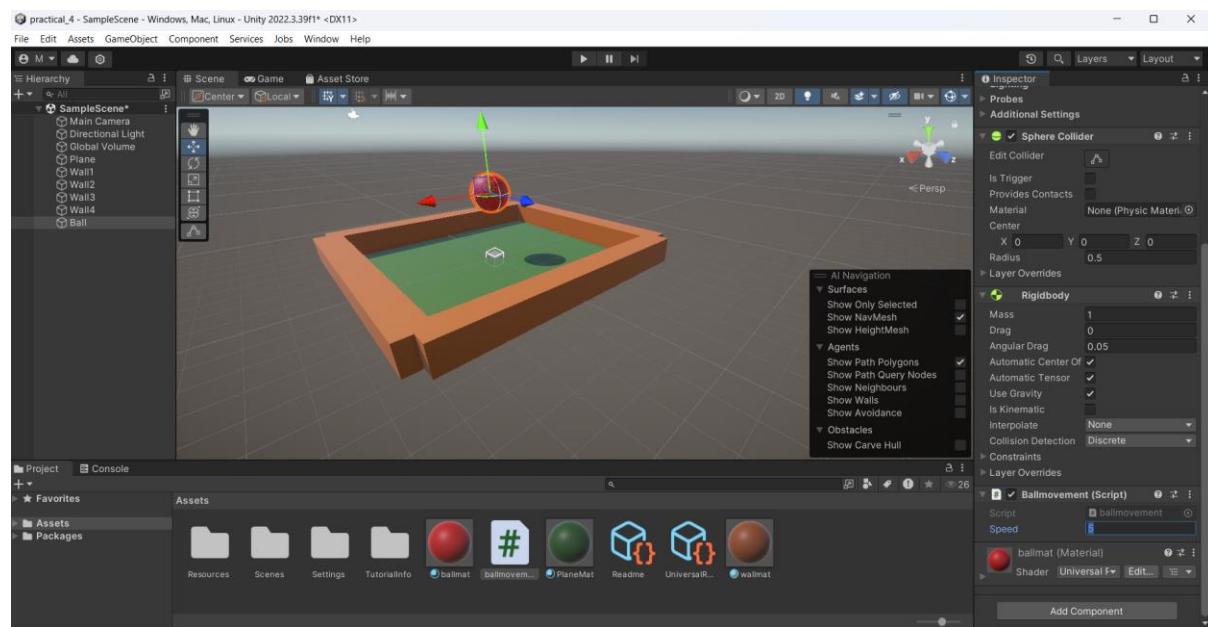
        float HorizontalMovement = Input.GetAxis("Horizontal");

        float VerticalMovement = Input.GetAxis("Vertical");

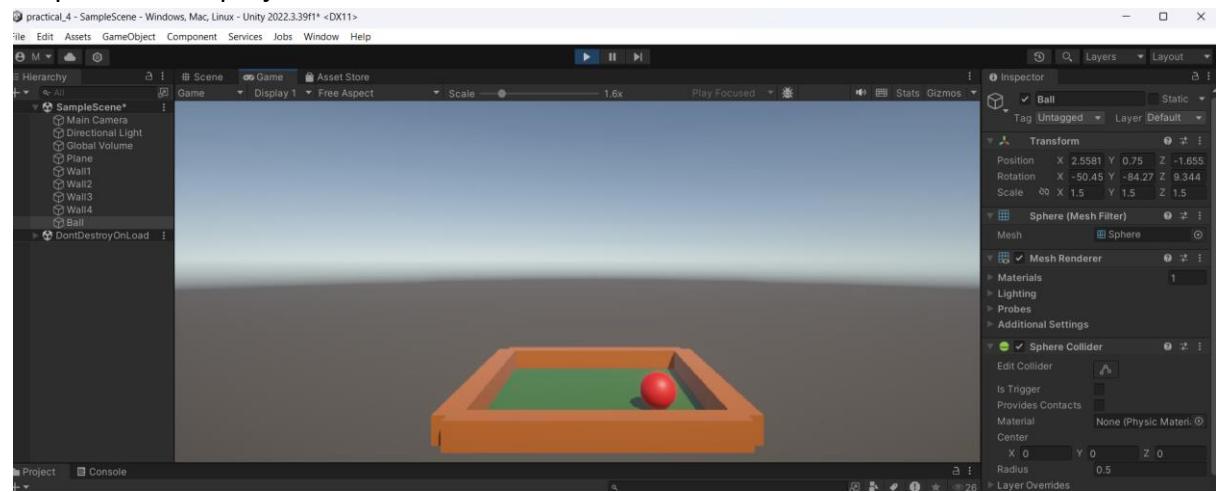
        Vector3 MoveBall = new Vector3(HorizontalMovement, 0, VerticalMovement);

        gameObject.transform.GetComponent<Rigidbody>().AddForce(MoveBall * Speed);

    }
}
```



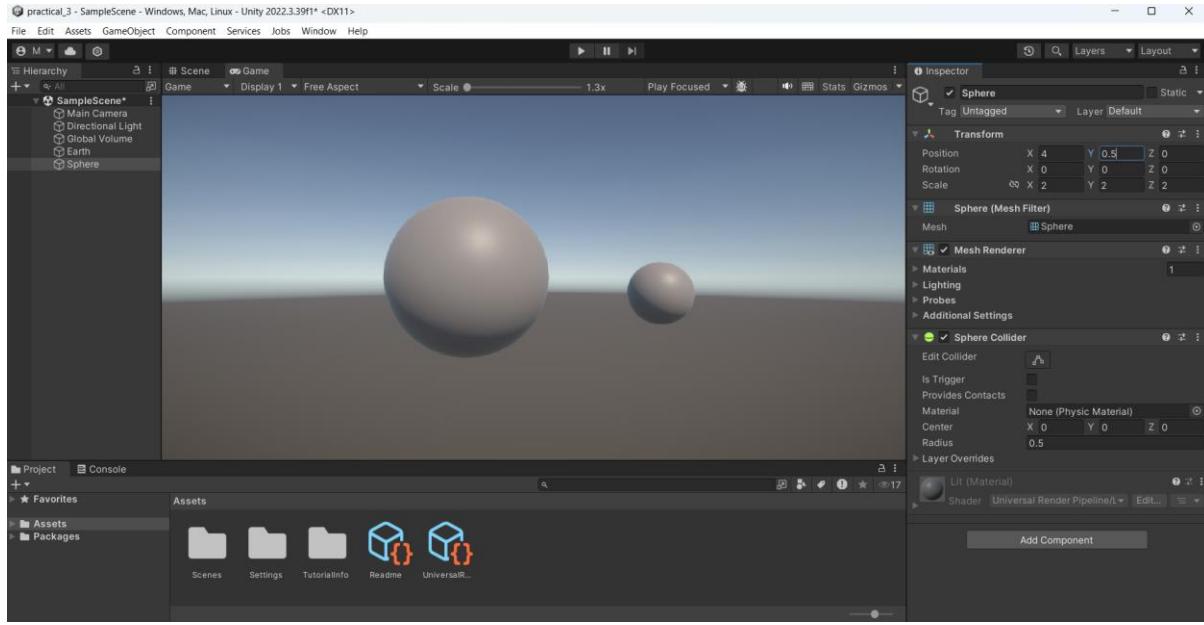
Step 7: Run the project.



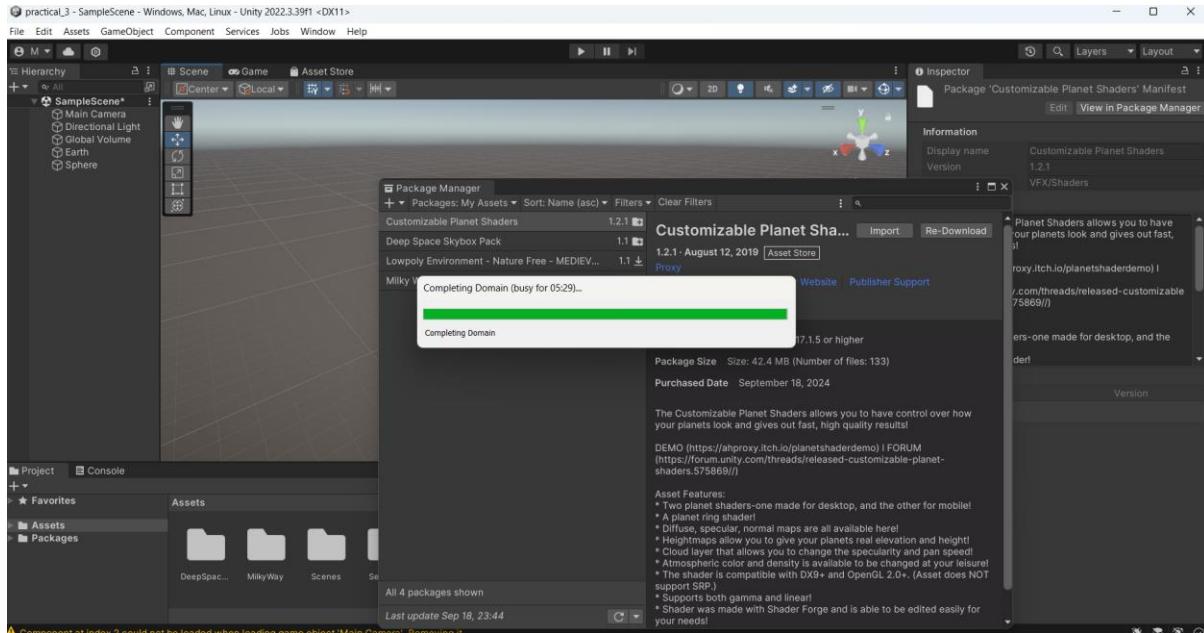
Practical 5: Add audio source.

Step 1: Open unity, create a new project.

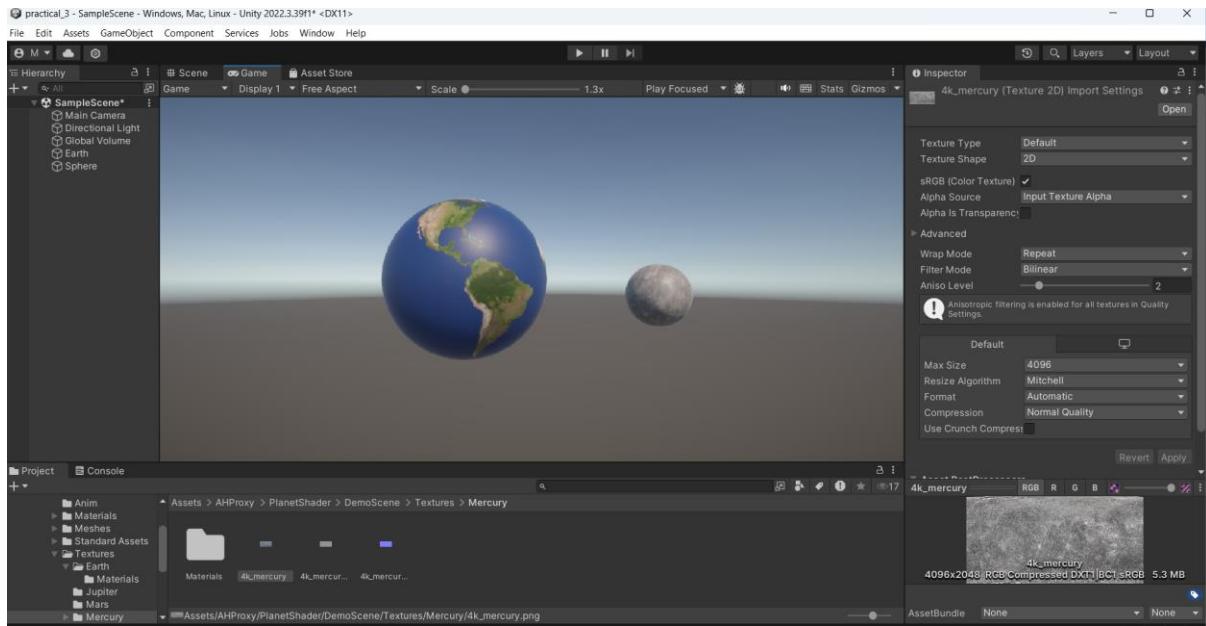
Step 2: Gameobject -> 3d object -> Sphere -> Change transform and size -> Take another sphere change transform and size.



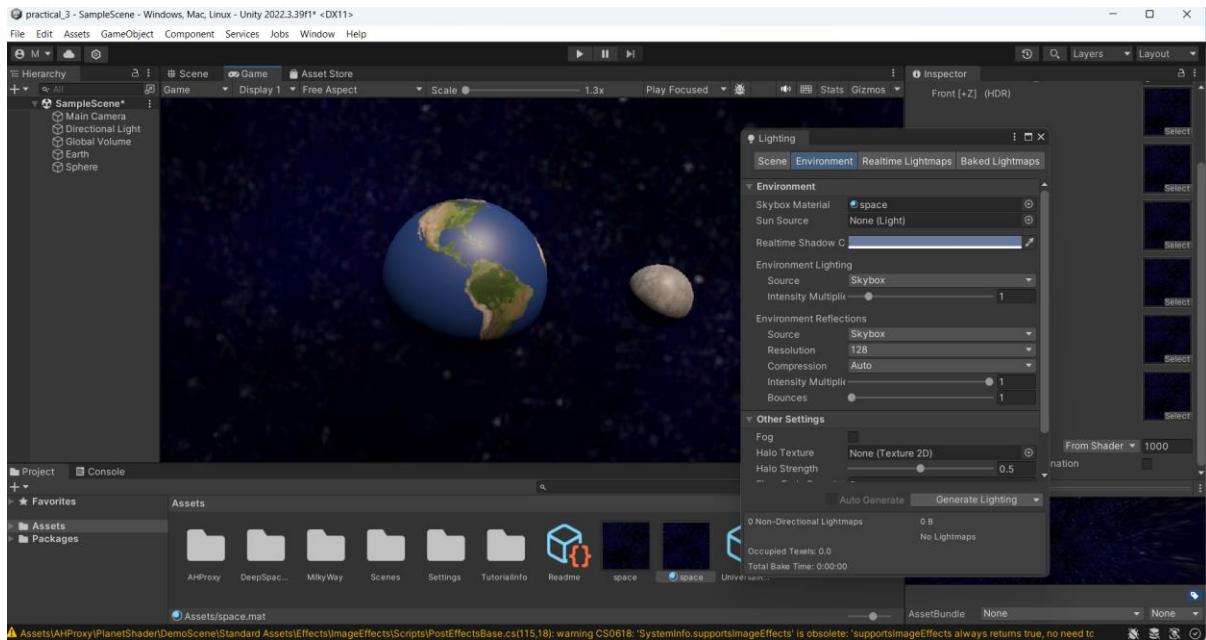
Step 3: Go to unity assets store download packages of textures that you want to apply on object and import it into project.



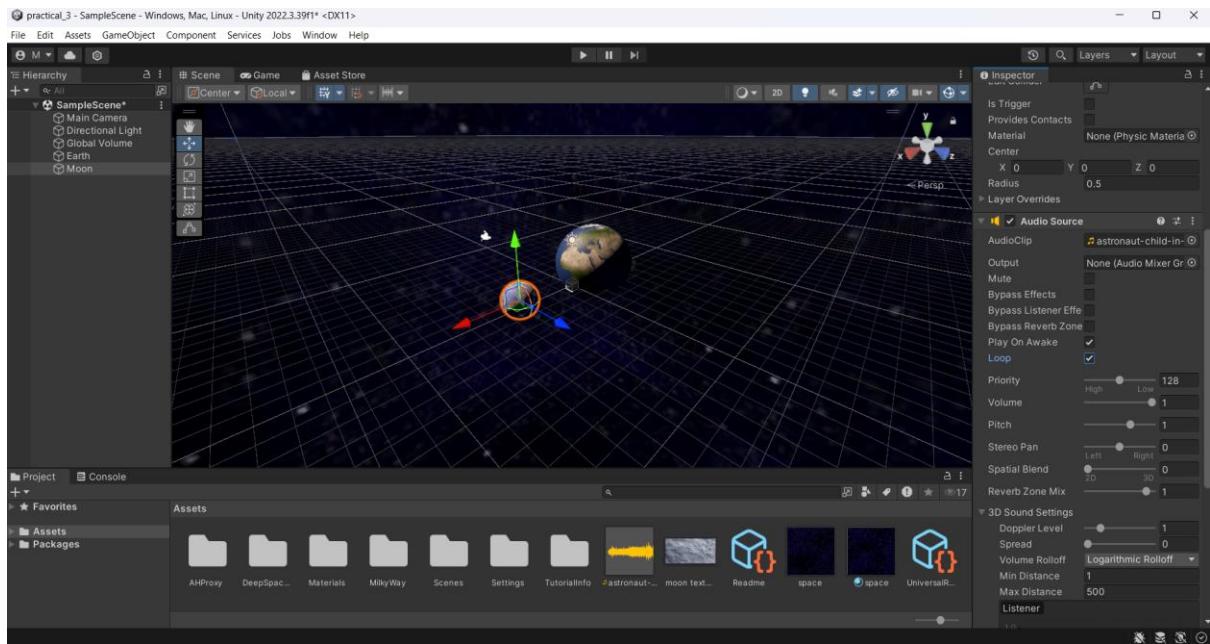
Step 4: Drag the Earth texture from assets on big sphere as well as moon texture on small sphere.



Step 5: Create material in assets name as space. Change shader as skybox in inspector panel. Window -> Rendering -> Environment -> Drag created space material on skybox material.



Step 6: import audio clip from file manager to assets -> Drag that sound on any object and click loop in inspector panel of object that you add audio source.



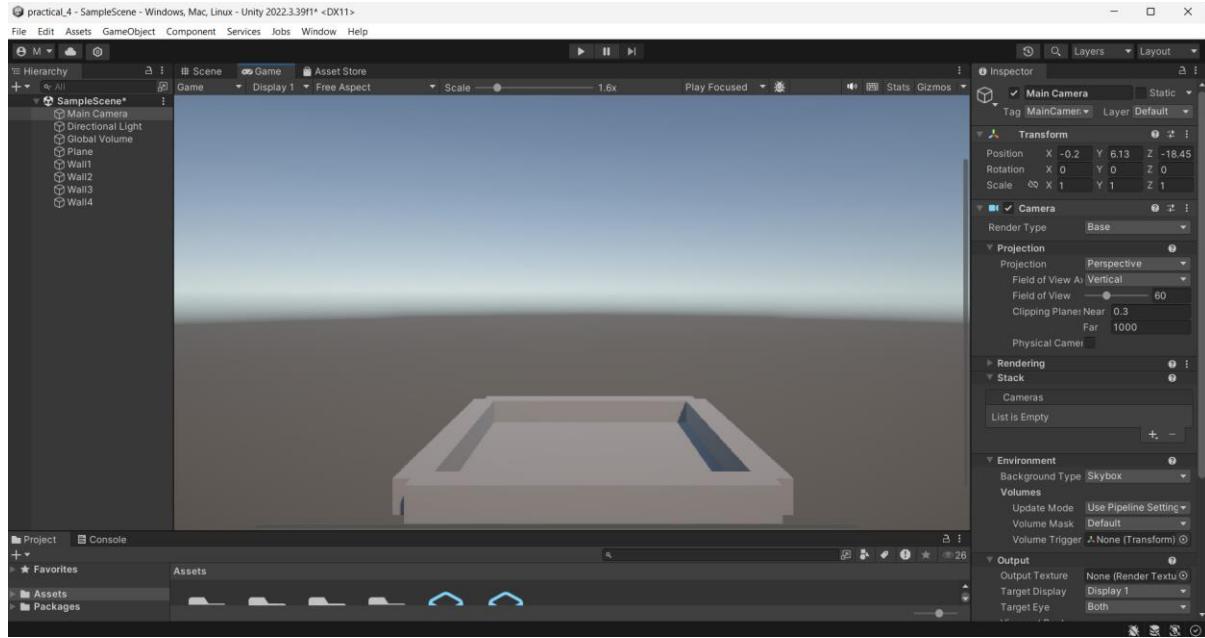
Step 7: Run the project.



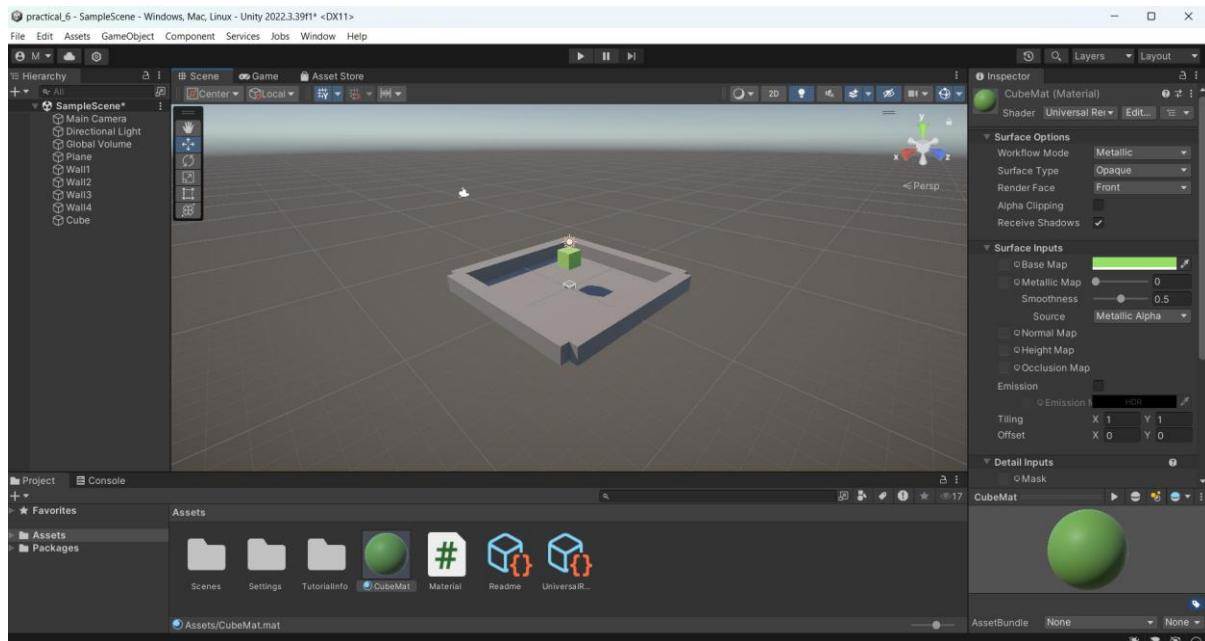
Practical 6: Changing colour of object using collision.

Step 1: Open unity, create a new project.

Step 2: Gameobject -> 3d object -> plane and change the transformation -> 3d object -> take 4 cubes and change the transformation.



Step 3: Take any 3d object for collision. Add Component “Rigidbody” on inspector panel of cube. Check gravity box. Create new material -> rename as cube mat -> colour it. Drag that material on cube.



Step 4: Right click on assets -> create -> c# script -> rename it and add code in it. Drag the script on each wall as well as on plane.

```
using System.Collections;
```

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

```
using UnityEngine;
```

```
public class Material : MonoBehaviour
```

```
{
```

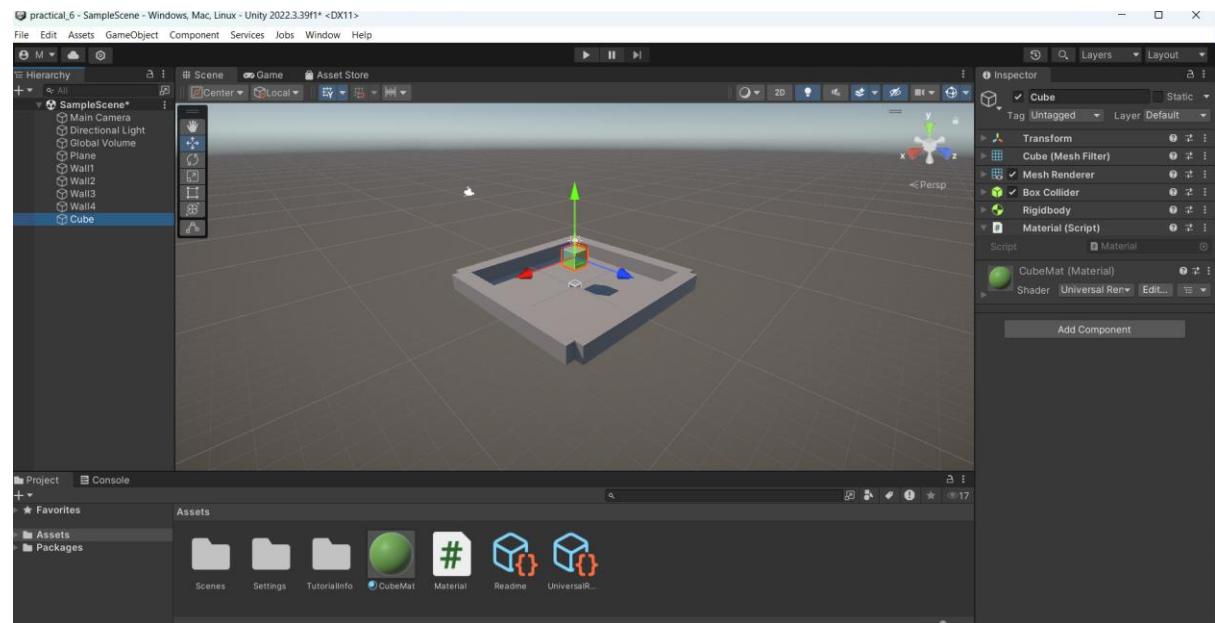
```
    private void OnCollisionEnter(Collision other)
```

```
{
```

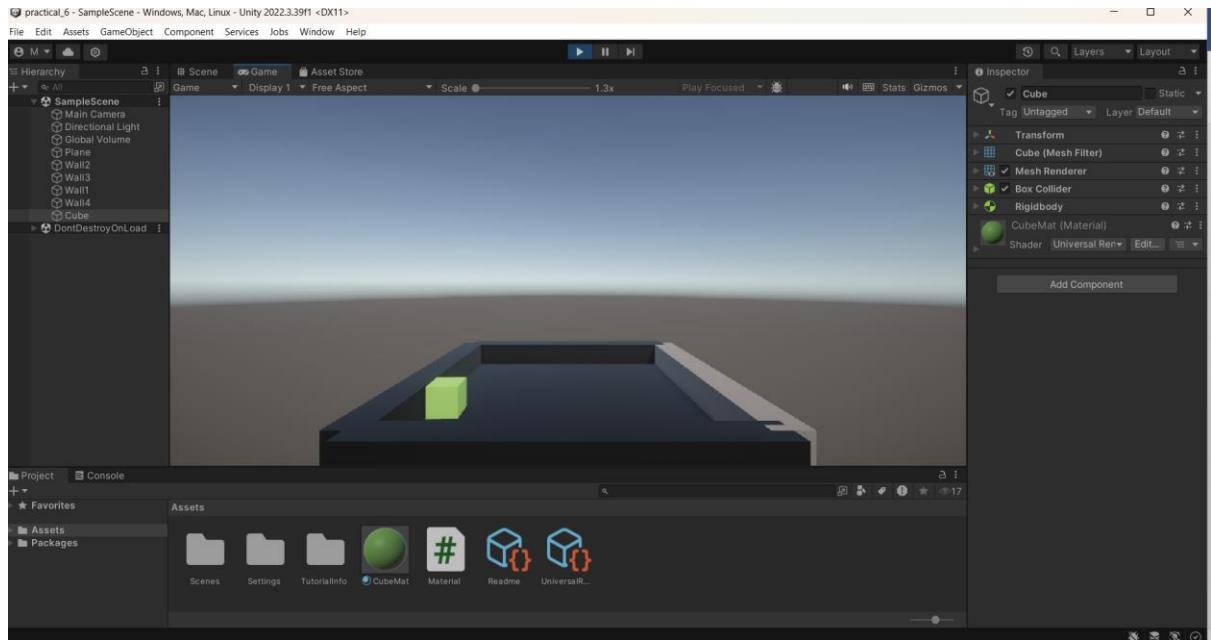
```
        GetComponent<MeshRenderer>().material.color = Color.black;
```

```
}
```

```
}
```



Step 5: Run the project.



Practical 7: Create a scene using terrain.

Step 1: Open Unity Hub software and Login

Step 2: Create New Project follow the steps 1. Click New project 2. Select 3D and Enter Project Name 3. Click Create Project

Step 3: Download Grass, Sand, and Water textures links given below.

1. Grass texture 2. Sand texture 3. Water texture

1. https://www.google.com/imgres?q=grass%20texture&imgurl=https%3A%2F%2Fimg.freepik.com%2Ffree-vector%2Fseamless-green-grass-pattern_1284-52275.jpg%3Fsize%3D338%26ext%3Djpg%26ga%3DGA1.1.2008272138.1724198400%26semt%3Dais_hybrid&imgrefurl=https%3A%2F%2Fwww.freepik.com%2Ffree-photos-vectors%2Fgrasstexture&docid=IYOkCw5WpN4uAM&tbnid=KeY24M1hVLOH1M&vet=12ahUKEwjK28uK1laIAxVYTGwGHe7QDP4QM3oECGoQAA..i&w=338&h=338&hcb=2&ved=2ahUKEwjK28uK1laAxVYTGwGHe7QDP4QM3oECGoQAA

2. <https://www.google.com/imgres?q=sand%20texture&imgurl=https%3A%2F%2Fi.pinimg.com%2F564x%2F96%2F86%2F80%2F968680bd14a9a60d02aae7cb0b1b8028.jpg&imgrefurl=https%3A%2F%2Fwww.pinterest.com%2Fpin%2Fsand-texture-seamless-free-photo-on-pixabay--656962664409568804%2F&docid=KEbsYJIIHglJJM&tbnid=dkXhfxJVfBj4dM&vet=12ahUKEwjwtNa01laIAxViSWwGHXaqDEEQM3oECFcQAA..i&w=507&h=338&hcb=2&ved=2ahUKEwjwtNa01laIAxViSWwGHXaqDEEQM3oECFcQAA>

3. https://www.google.com/imgres?q=water%20texture&imgurl=https%3A%2F%2Fmedia.istockphoto.com%2Fid%2F96336795%2Fphoto%2Fa-close-up-image-of-a-hottub-with-blue-water.webp%3Fb%3D1%26s%3D170667a%26w%3D0%26k%3D20%26c%3D5xAybv5svmlgrD9Vaplc8PjbnXwZl3lV1mruXlwc%3D&imgrefurl=https%3A%2F%2Funsplash.com%2Fs%2Fphotos%2Fwatertexture&docid=ziWS1ebeTNH3_M&tbnid=mAvzzw3t14iwoM&vet=12ahUKEjk15rE1laIAxUWUGcHHbQfEAYQM3oECHUQAA..i&w=508&h=340&hcb=2&ved=2ahUKEjk15rE1laIAxUWUGcHHbQfEAYQM3oECHUQAA

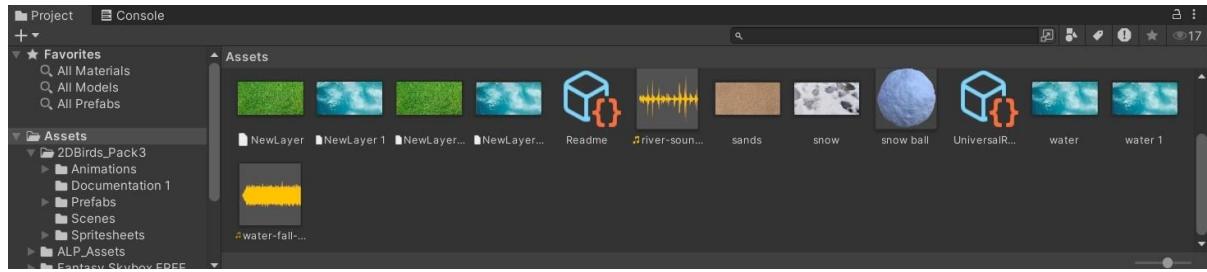
Step 4: Download sounds effect for birds, river, and waterfalls website download link below

1. Birds: <https://pixabay.com/sound-effects/search/birds/>

2. River: <https://pixabay.com/sound-effects/search/river/>

3. Waterfall: <https://pixabay.com/sound-effects/search/waterfall/>

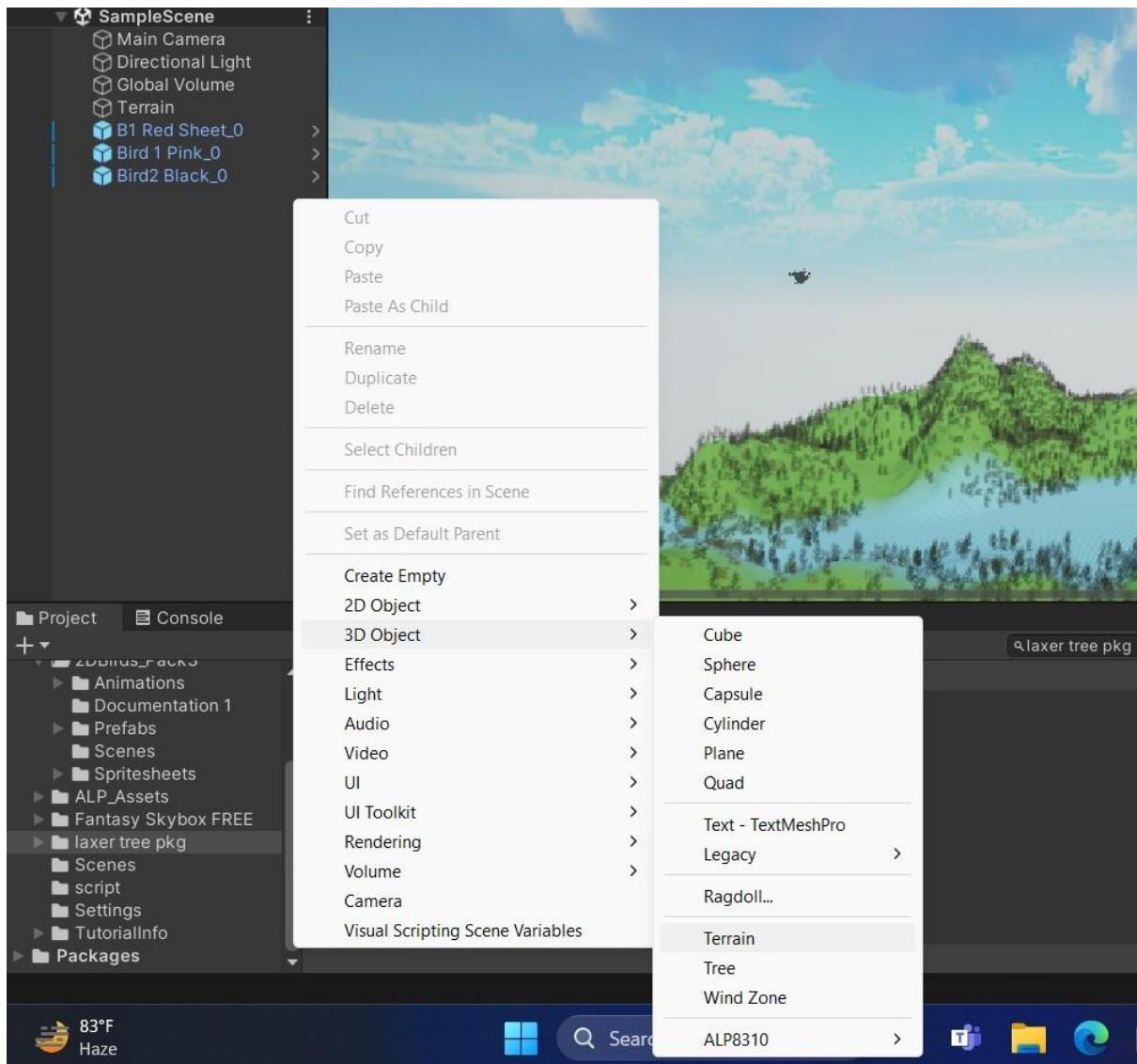
Step 5: Import all downloaded material on your unity hub assets folder. 1. Right click on assets folder and then click Import New Assets. 2. Select your downloaded files and sounds and click import 1. 2. 3.



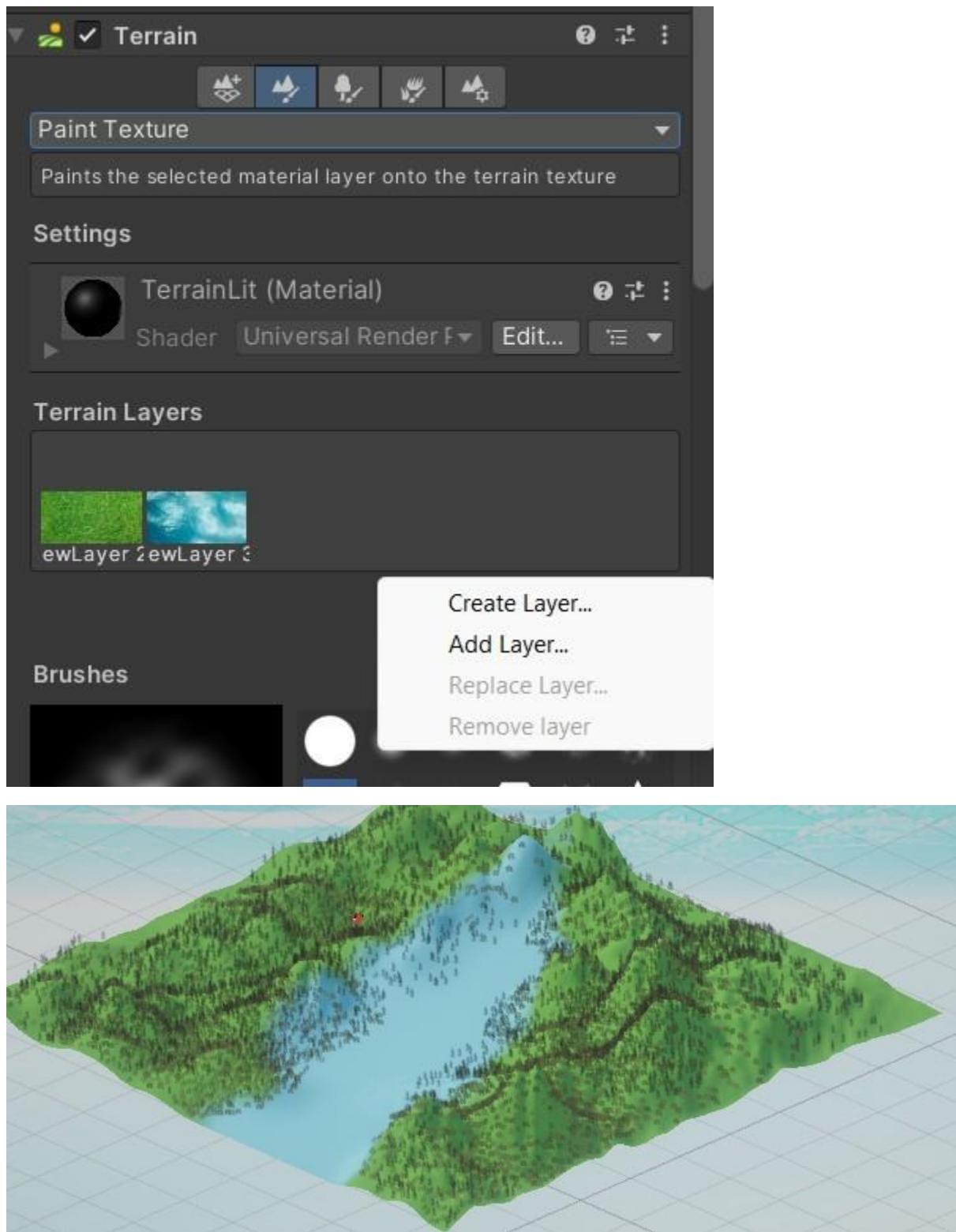
Step 6: Download and import unity asset store. Add to my store

1. 2D Cartoon Birds pack <https://assetstore.unity.com/packages/2d/characters/2dcartoon-birds-pack-149167> . 2. Mobile Tree Package. 3. Click Add to My Assets. 4. Click Open to Unity. 5. Click Download and import. 6. Click Import than click Ok

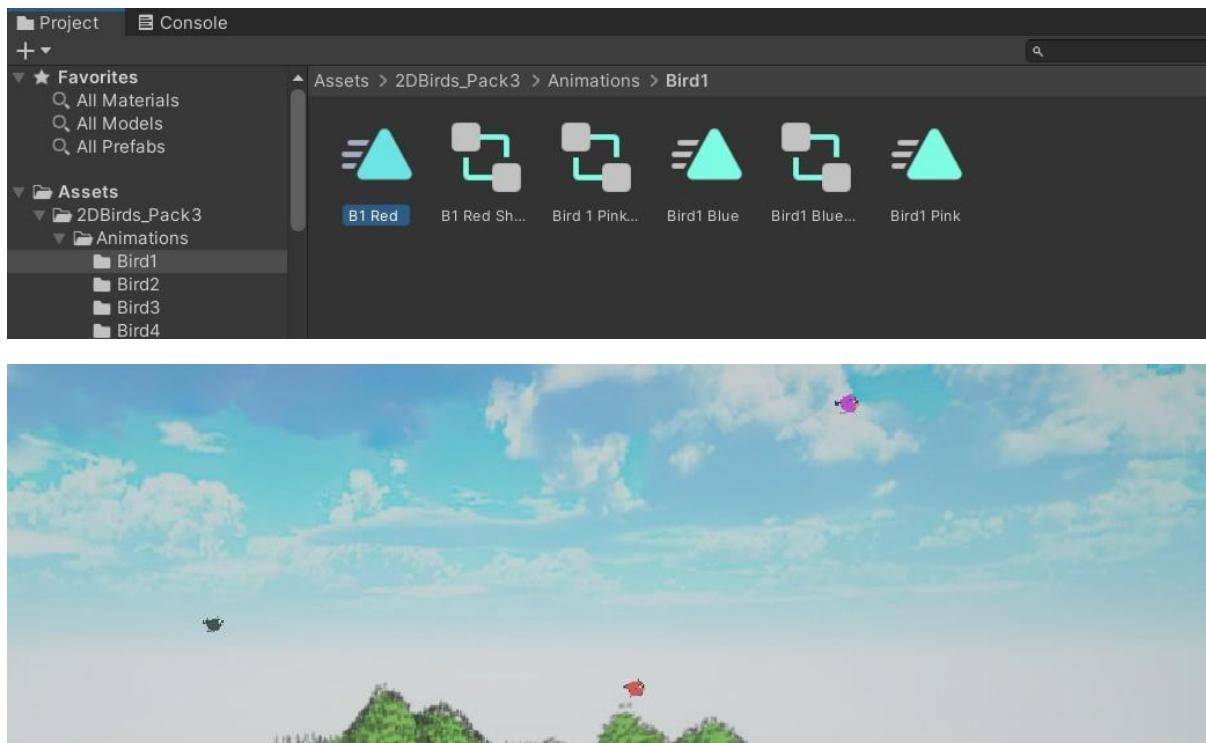
Step 7: Create a Terrain on my project. Left side in Sample Scene click Right click than select 3D Object and then select a Terrain.



Step 8: Add terrain Layers 1. Click Paint Terrain 2. Click Edit Terrain Layers and then click Create Layers and select your terrain images 3. Select Raise or Lower Terrain and then create mountains and river using below multiples brush. 4. Add Trees on mountains. Click on Paint Trees than click on Edit Trees and Add Tree. Click Game Object and select your trees. 5. Create your trees on mountain. Select your tree and then past on mountain.



Step 9: Add birds on mountain. Go to Assets and then select a 2DBirds_Pack3 and click Animations and select Red, Black, Pink birds. Selects Birds and increase size of birds.



Step 10: Write a C# script for move birds. Create a script folder than wire a C# code below.
 Filename: BirdMovement

```
Code: using System.Collections; using System.Collections.Generic; using UnityEngine; public
class BirdMovement : MonoBehaviour { public float speed = 5f; public Vector3 startPosition;
public Vector3 endPosition; void Start() { transform.position = startPosition; } void Update() {
transform.position = Vector3.MoveTowards(transform.position,endPosition,speed*Time.deltaTime);
if (transform.position == endPosition) { transform.position = startPosition; } } }
```

Step 11: Add downloaded audio (Water Fall, Birds, Rever). 1. Click Terrains 2. Right side panel click Add Components 3. Right side panel click Add Components 4. Click on Audio 5. Click on Audio Source 6. Click on None Audio clip and select your downloaded (Birds, Waterfall, Rever) audio files.

Inspector

Terrain Collider

Provides Contacts

Material None (Physic Material)

Terrain Data New Terrain 1

Enable Tree Collider:

► Layer Overrides

Audio Source

AudioClip birds-singing_nature-sound-1

Output None (Audio Mixer Group)

Mute

Bypass Effects

Bypass Listener Effe

Bypass Reverb Zone

Play On Awake

Loop

Priority 128

Volume 1

Pitch 1

Stereo Pan 0

Spatial Blend 0

Reverb Zone Mix 1

► 3D Sound Settings

Audio Source

AudioClip water-fall-smooth-relaxing-1:

Output None (Audio Mixer Group)

Mute

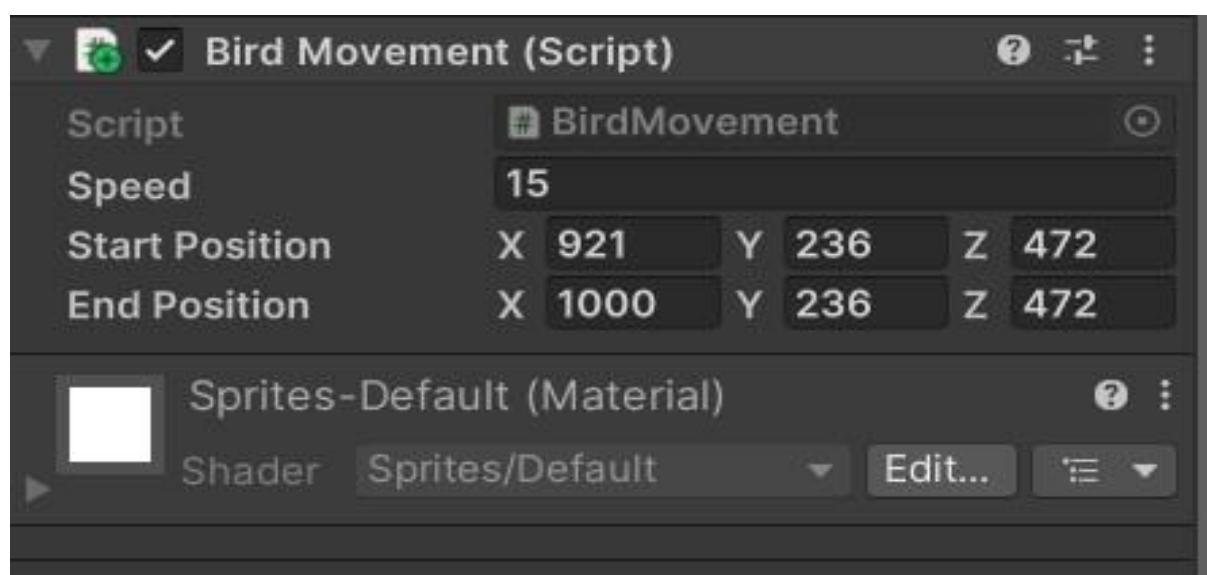
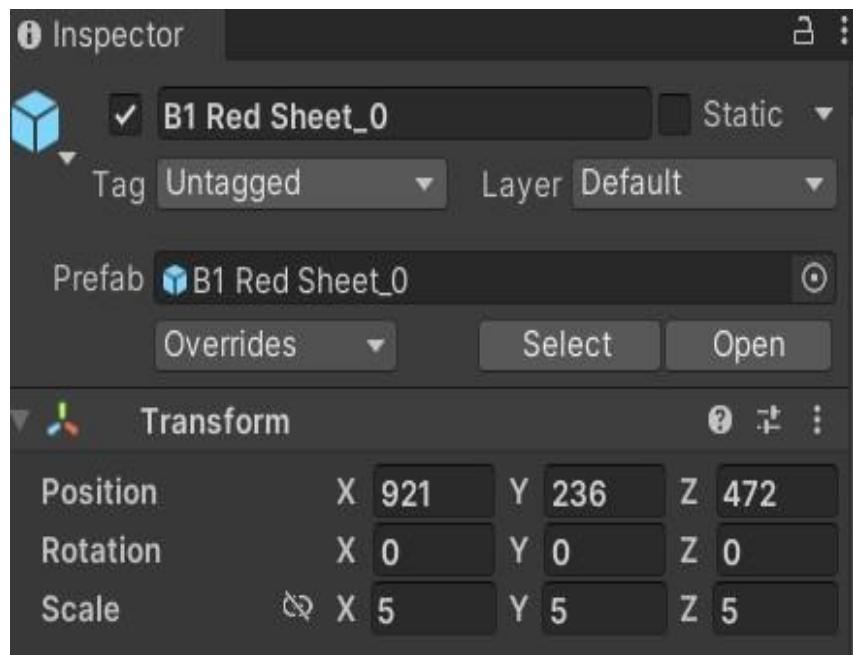
Bypass Effects

Bypass Listener Effe

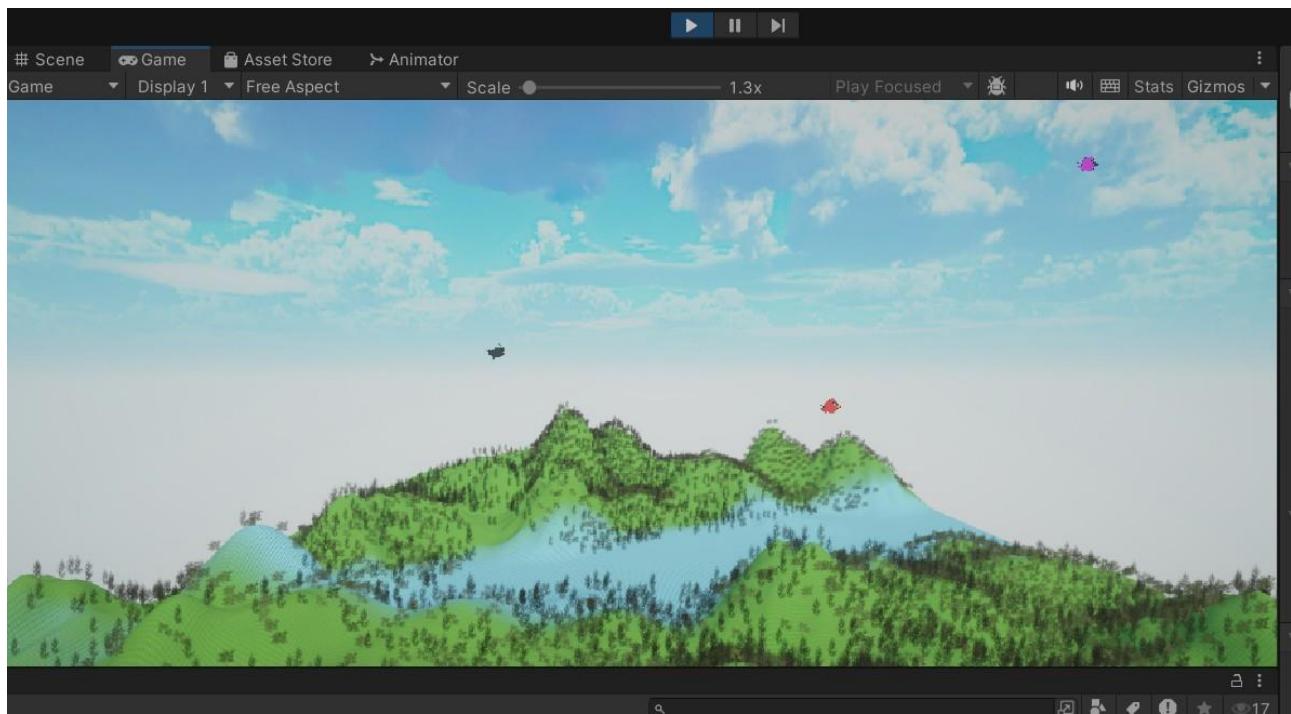
Bypass Reverb Zone

Play On Awake

Step 12: Set the Birds starting and Ending position or bird speeds. Select every bird (Red, Black, Pink) and set the positions. Red Bird: Same setting for every birds. Scroll down and set the Bird Movement (Script).



Step 13: Run the unity project.



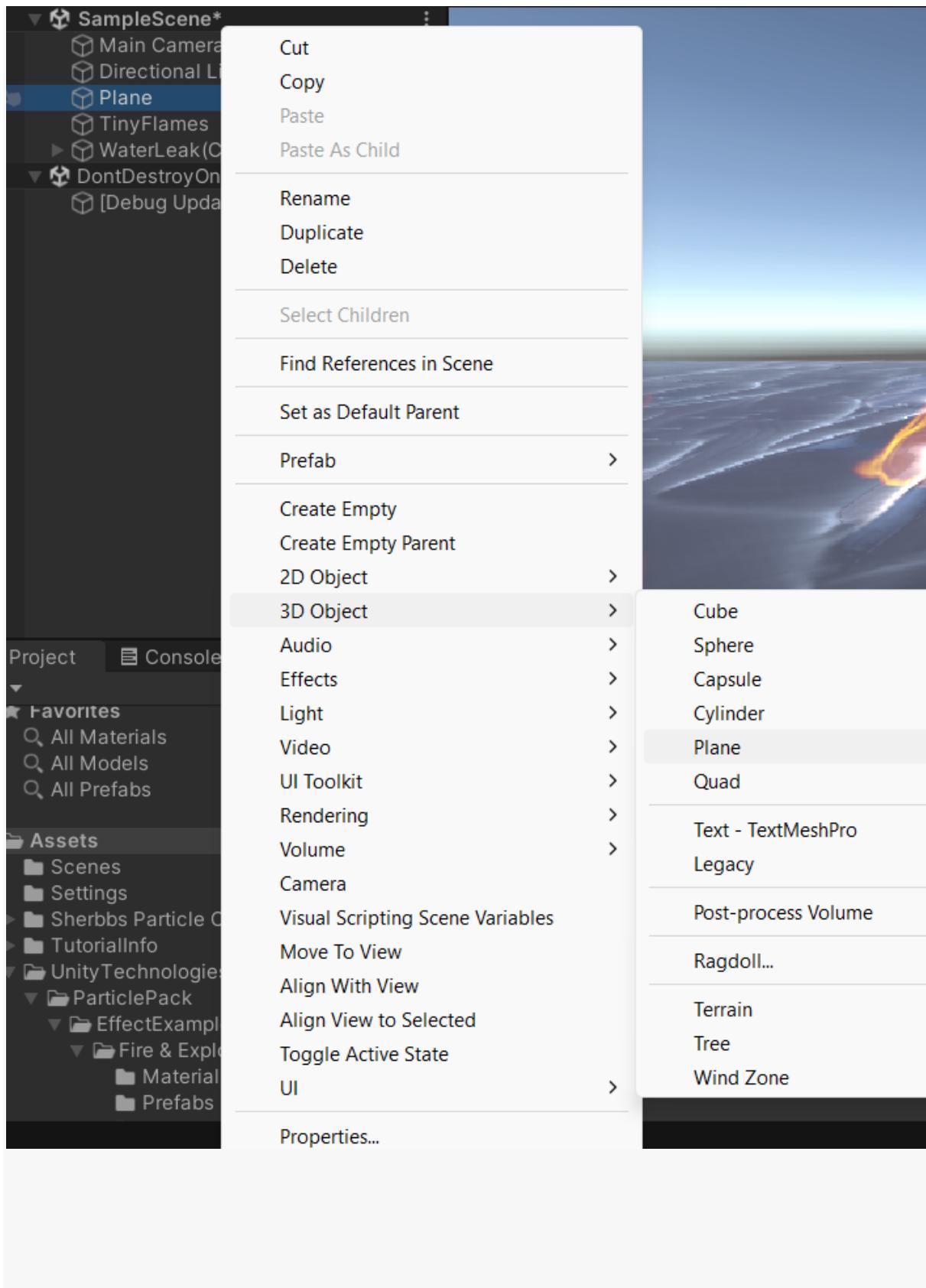
Practical 8: Create a scene using particle systems.

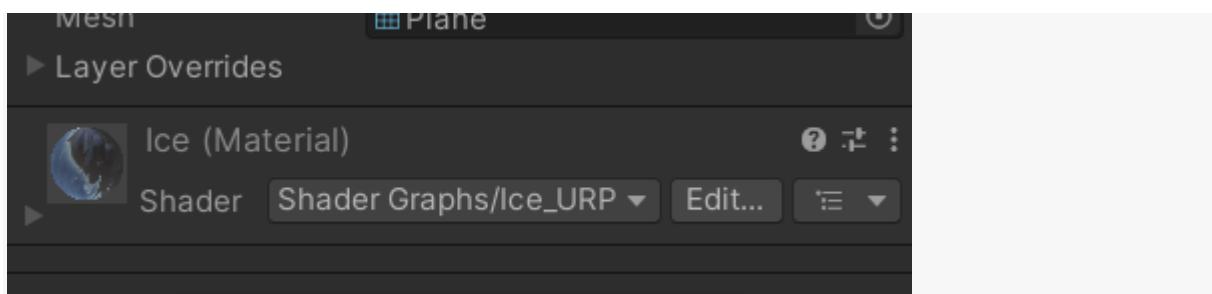
Open Unity, create new project.

Step 1: Set Up Your Unity Scene

1. Create a Plane:

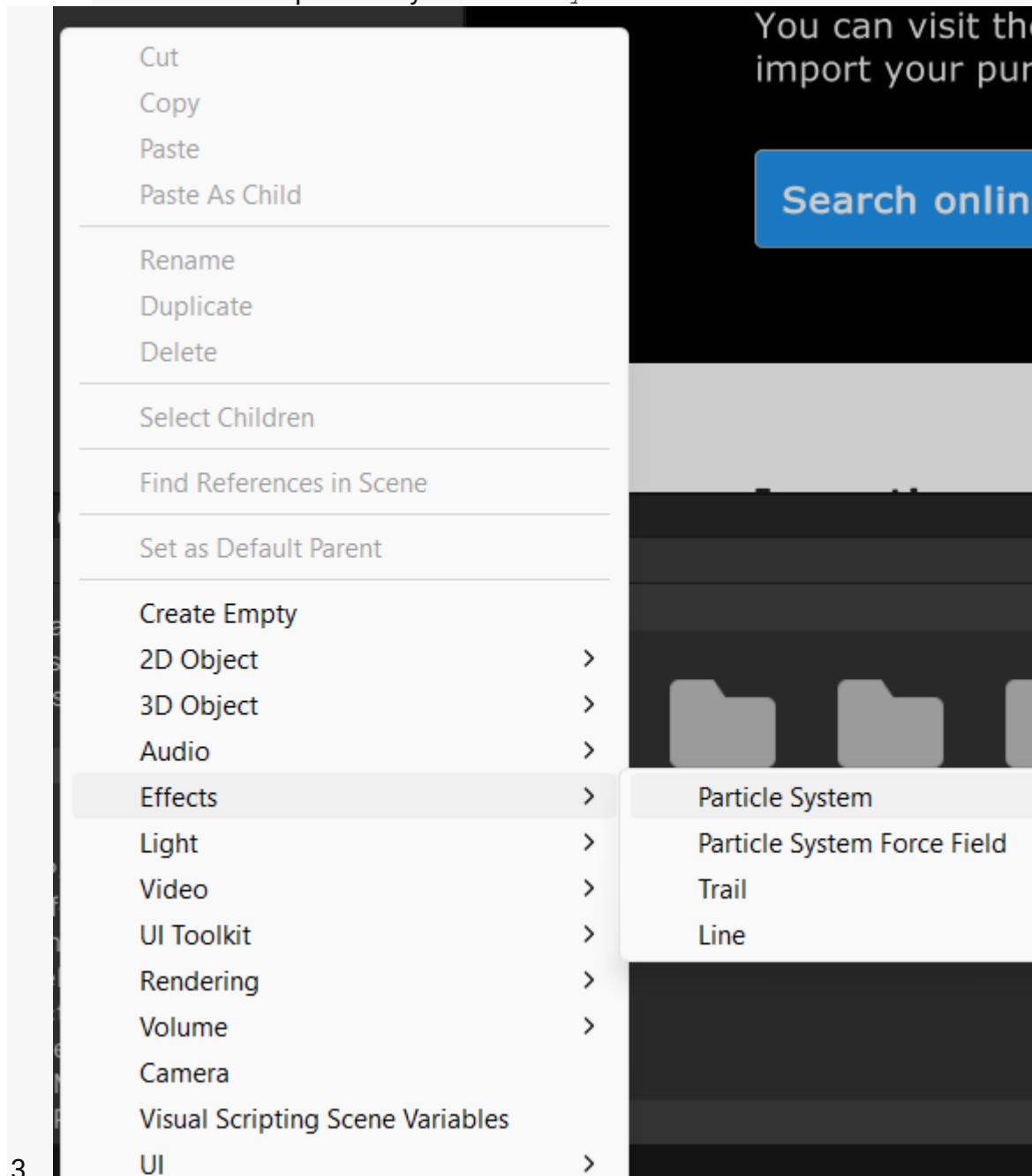
- Right-click in the **Hierarchy** window.
- Go to **3D Object > Plane**.
- Apply your ice texture to the plane by dragging the texture onto the plane in the **Scene** view or **Inspector**.





2. Add a Particle System:

- o Right-click in the **Hierarchy** window.
- o Go to **Effects > Particle System**.
- o Rename the particle system to `TinyFlames`.



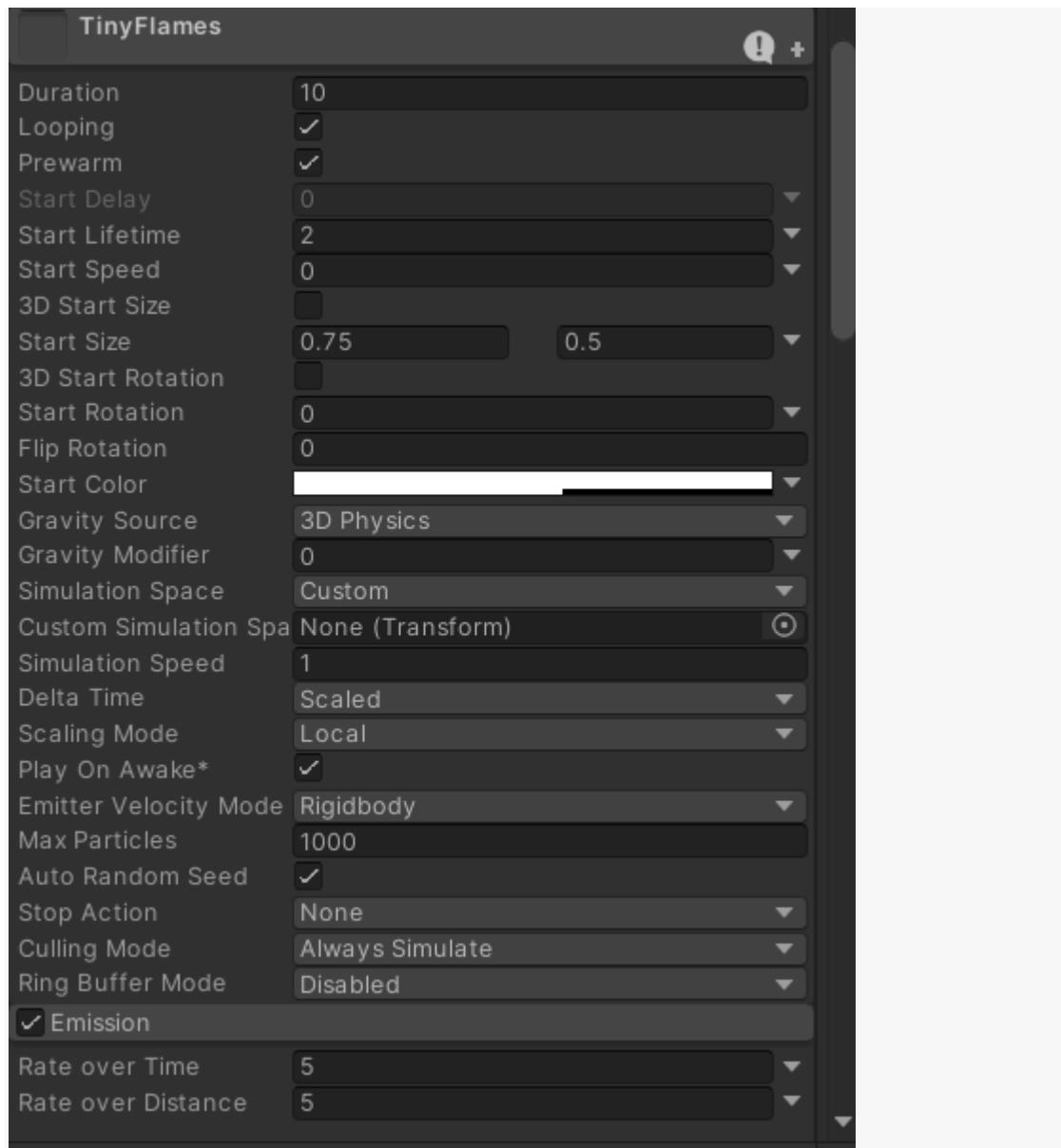
Step 2: Configure the Particle System

1. Select the Particle System:

- Click on `TinyFlames` in the Hierarchy to see its properties in the **Inspector** window.

2. Adjust Basic Settings:

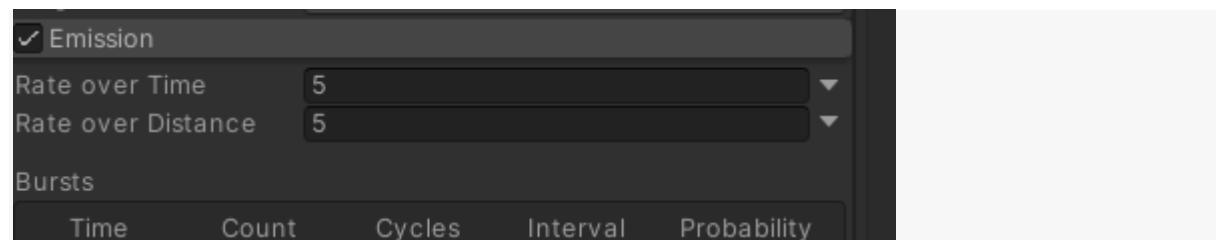
- **Duration:** Set to 10.
- **Looping:** Check this box to make the particle system loop.
- **Start Delay:** Set to 0.
- **Start Lifetime:** Set to Random between two constants with values 0.75 and 1.
- **Start Speed:** Set to Random between two constants with values 0.2 and 0.5.
- **Start Size:** Set to Random between two constants with values 0.5 and 1.
- **Start Rotation:** Set to Random between two constants with values 0 and 3.
- **Flip Rotation:** Set to 0.
- **Start Color:** Choose a color that represents flames (e.g., orange or red).



Step 3: Customize the Emission

1. Emission Module:

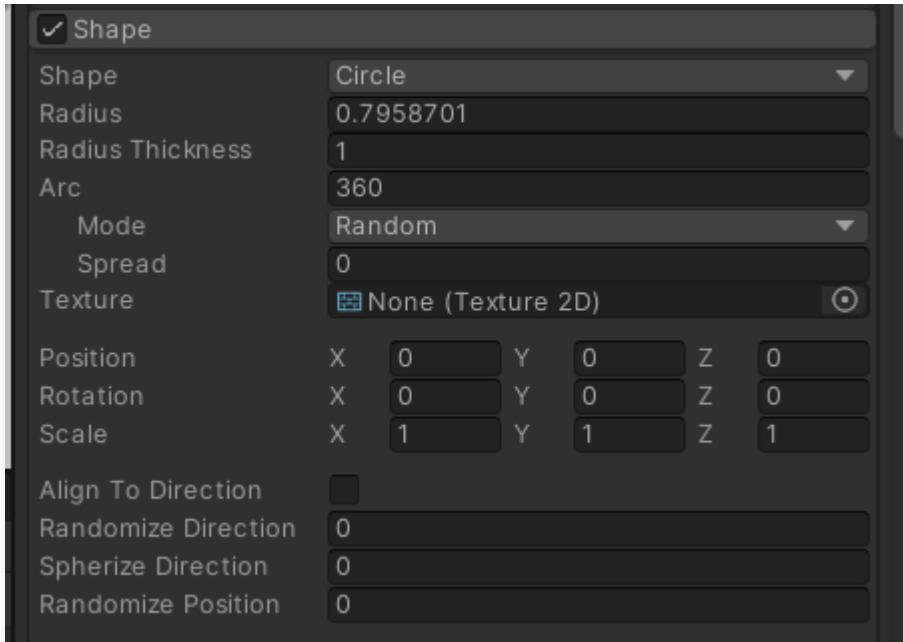
- Expand the **Emission** module in the Inspector.
- **Rate over Time:** Set to 10.
- **Rate over Distance:** Leave this disabled.



Step 4: Shape the Emission

1. Shape Module:

- Expand the **Shape** module.
- Choose the shape from which particles are emitted (e.g., cone, sphere, box). For flames, a cone shape is often effective.



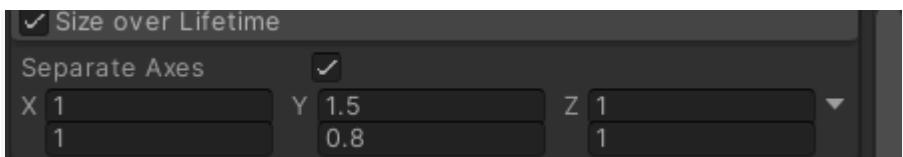
Step 5: Add Visual Effects

1. Color over Lifetime:

- Expand the **Color over Lifetime** module.
- Click the color bar to open the gradient editor and set the color changes over the particle's lifetime (e.g., from yellow to red).

2. Size over Lifetime:

- Expand the **Size over Lifetime** module.
- Check the box and adjust the curve to change the particle size over its lifetime.



Step 6: Add Movement and Rotation

1. Velocity over Lifetime:

- Expand the **Velocity over Lifetime** module.
- Adjust the values to give particles initial velocity.

2. Rotation over Lifetime:

- Expand the **Rotation over Lifetime** module.
- Check the box and adjust the curve to rotate particles over their lifetime.

Step 7: Test and Iterate

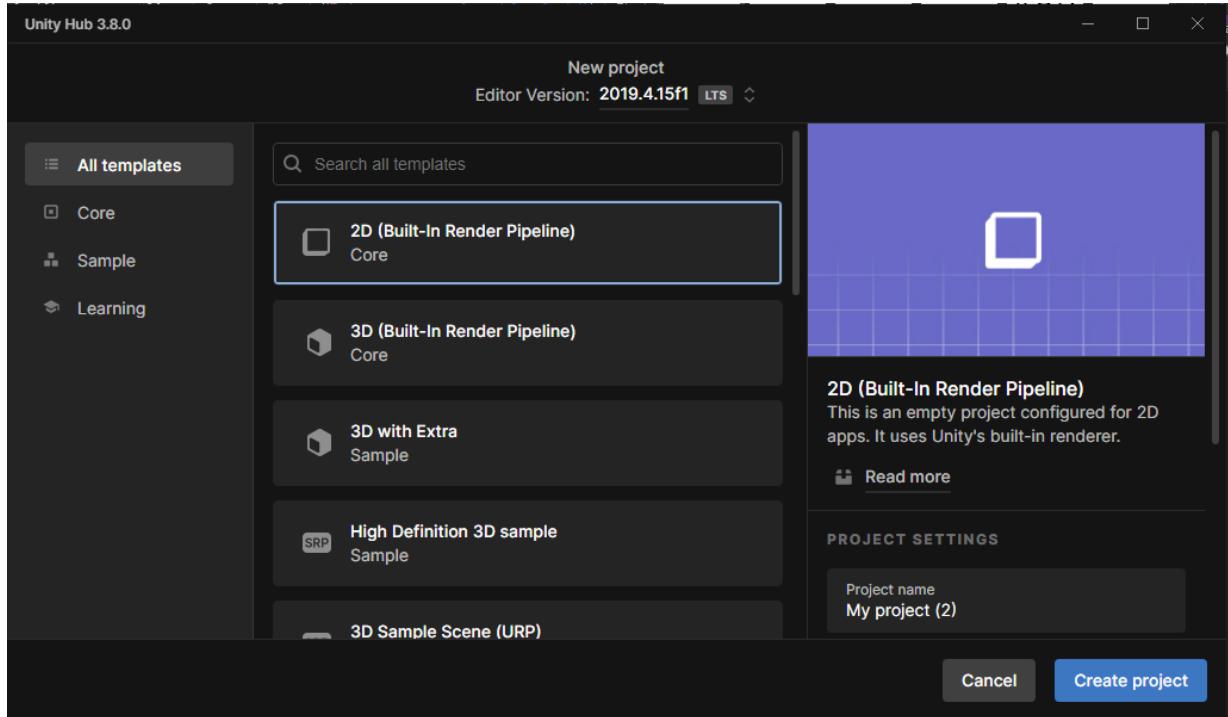
1. Play the Scene:

- Click the **Play** button at the top of the Unity editor to see your particle system in action.
- Adjust the settings as needed to achieve the desired effect.

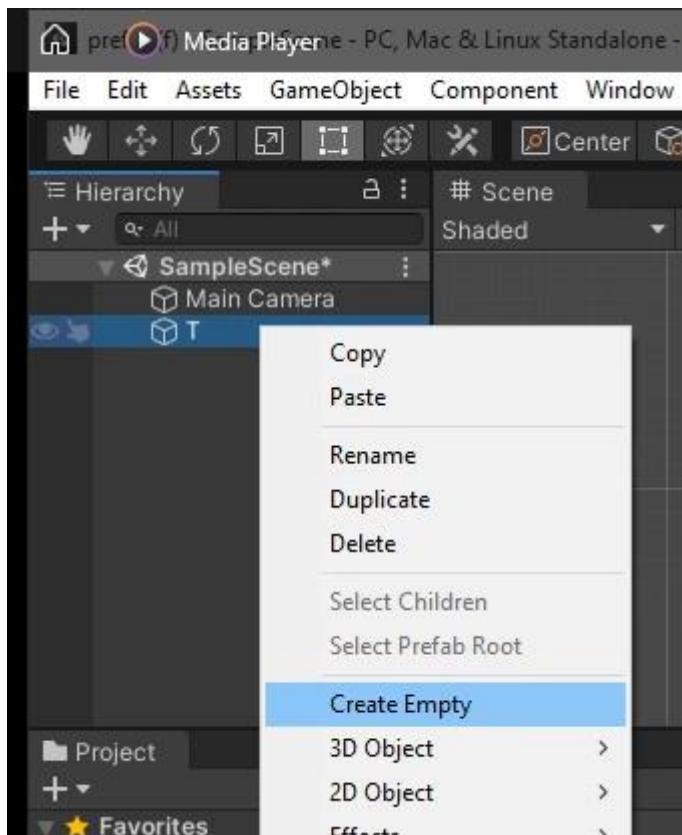


Practical 9: Create a scene using prefab.

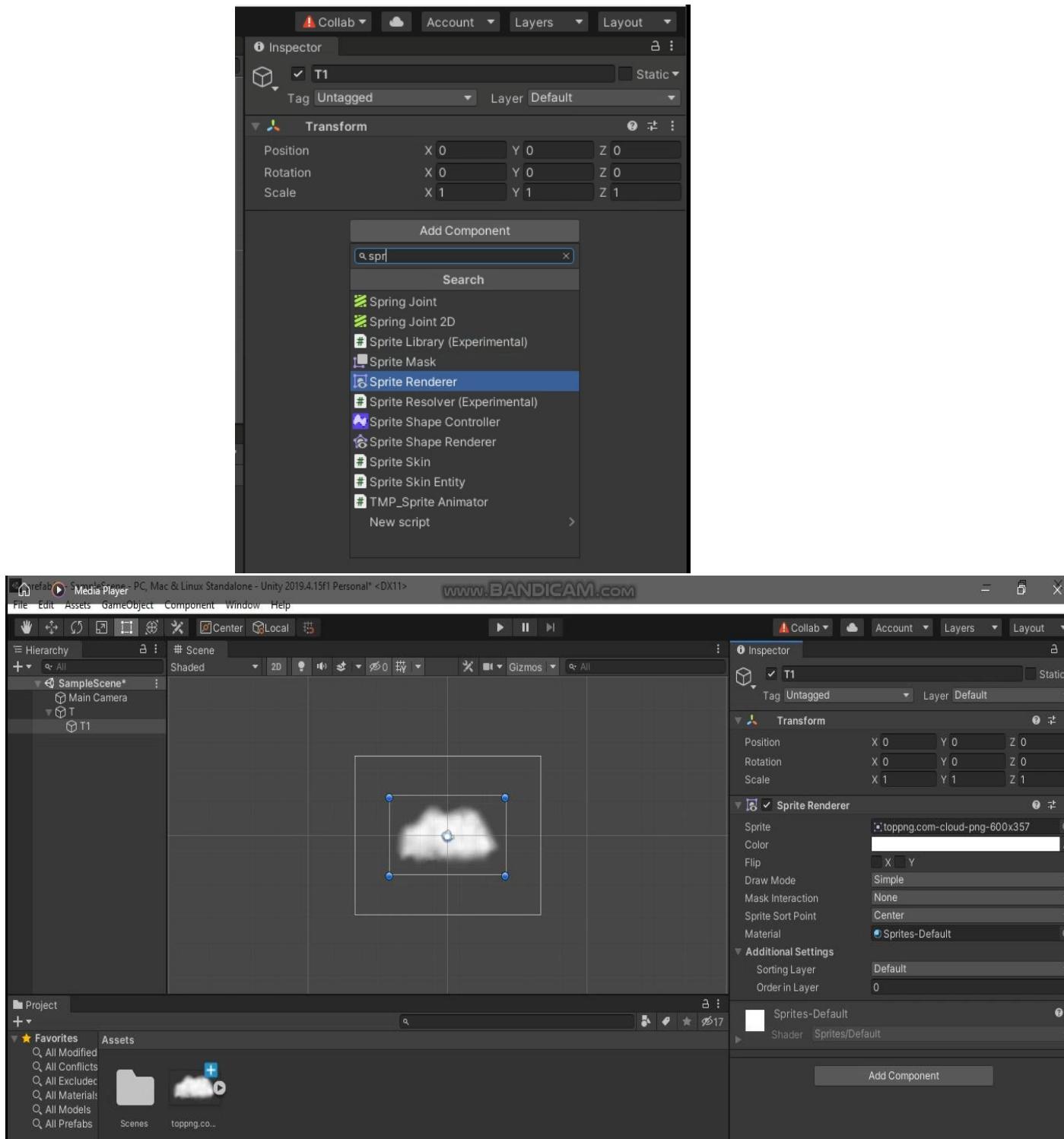
Step 1: Open Unity. Open a new 2D (Render Pipeline) Project in unity.



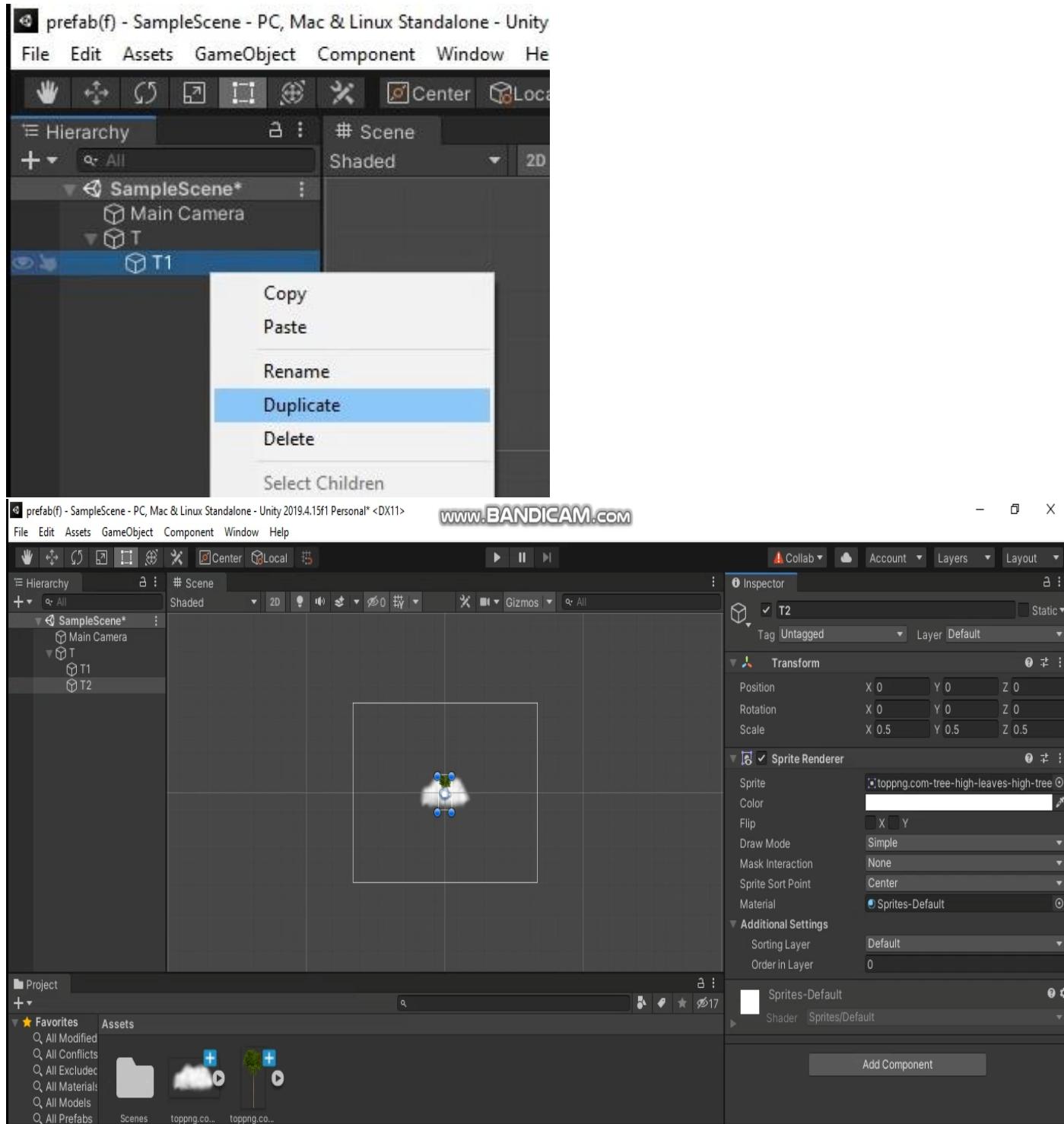
Step 2: Create a new Game Object, after that again create a new game object by right clicking on the existing game object in order to make a child game object.



Step 3: Add a new sprite render to the child game object to add image to it.



Step 4: Duplicate the child object to create the second game object change the image reposition the 2nd Child game object



Step 5: Select the Parent object, add a new Script and type the following code in that script:

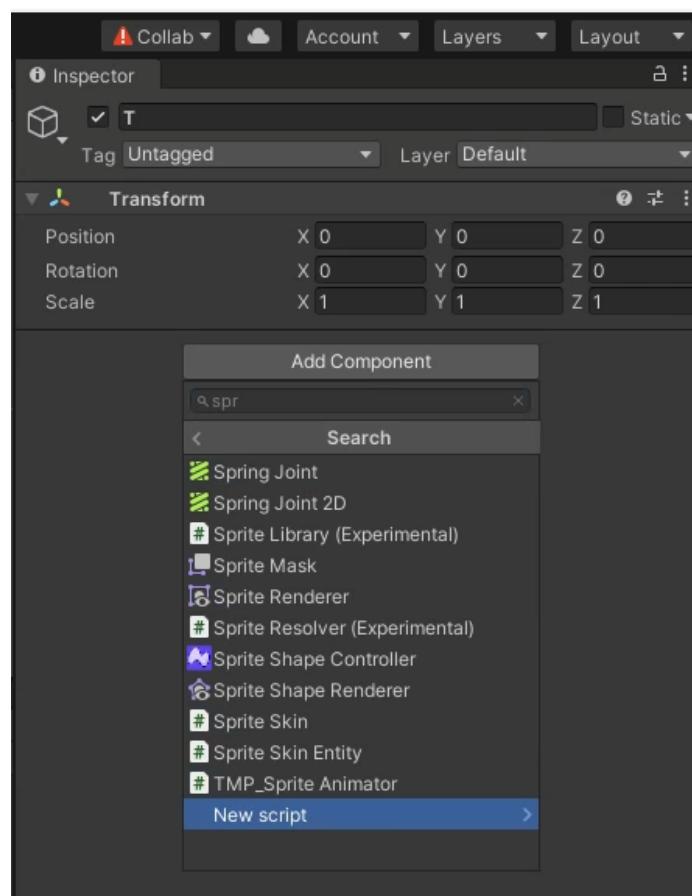
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class T : MonoBehaviour
{
    public float moveSpeed = 5;

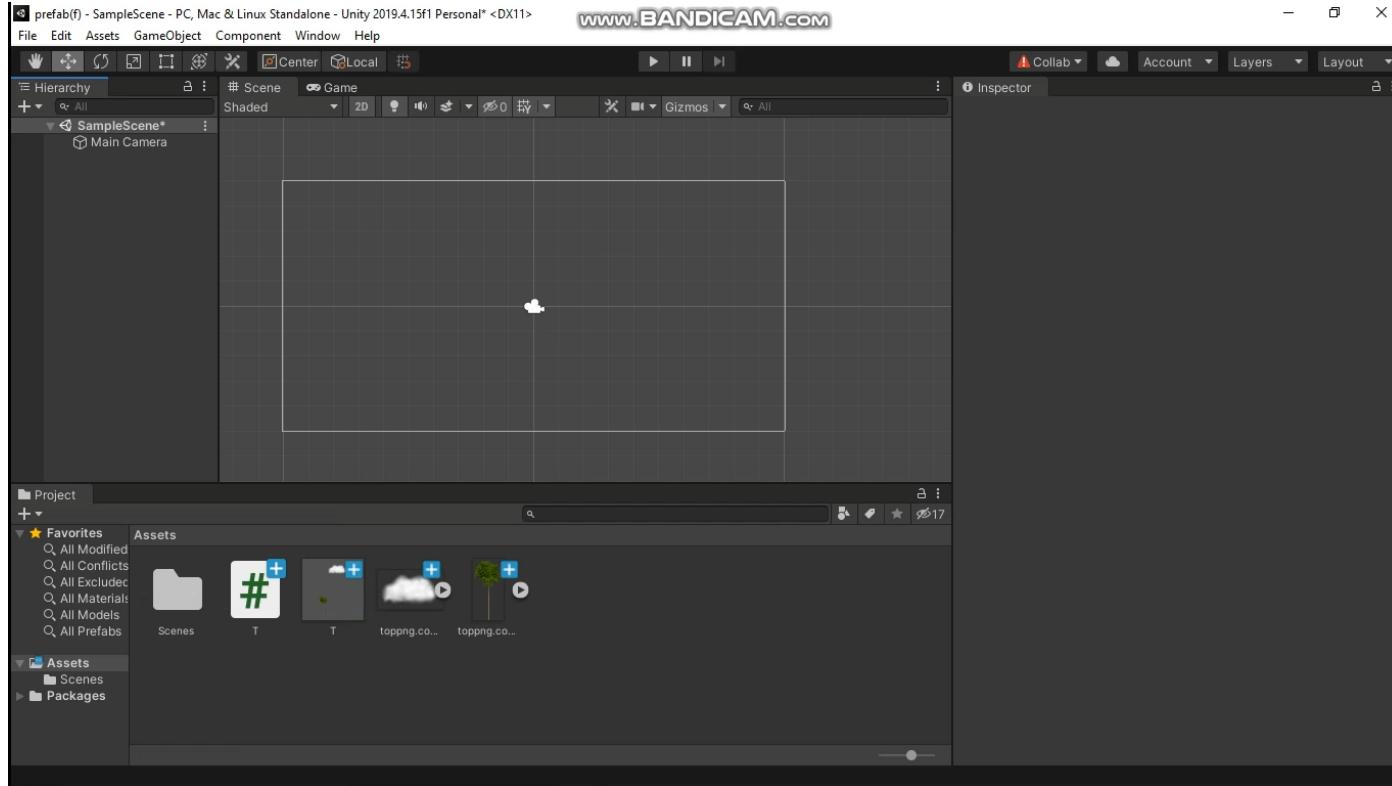
    // Start is called before the first frame update
    void Start()
    {

    }

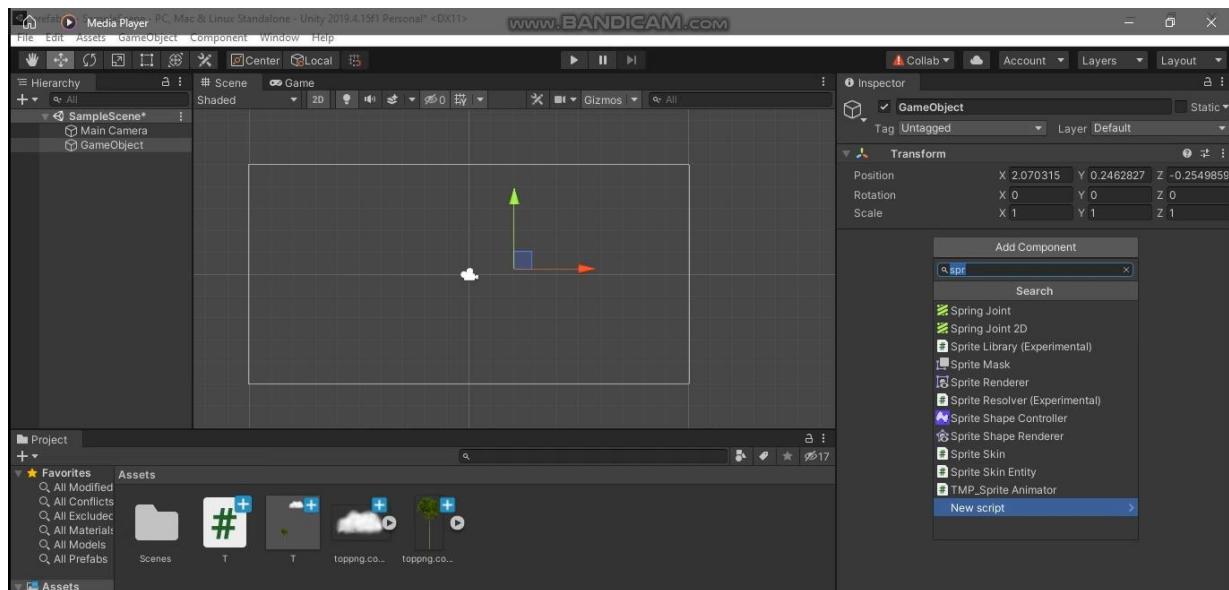
    // Update is called once per frame
    void Update()
    {
        transform.position = transform.position + (Vector3.left * moveSpeed) * Time.deltaTime;
    }
}
```



Step 6: Drag the parent Game Object in the Assets panel. This will create a pre-fab. Delete the Parent game Object in hierarchy Panel.



Step 7: Now Create a New Game Object add a new script to it and type the following code:



```
using System.Collections;
```

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

```
using UnityEngine;

public class TS : MonoBehaviour

{
    public GameObject T;
    public float spwanRate = 2;
    private float timer = 0;
    public float heightOffset = 0;

    // Start is called before the first frame update
    void Start()

    {
        spwanT();
    }

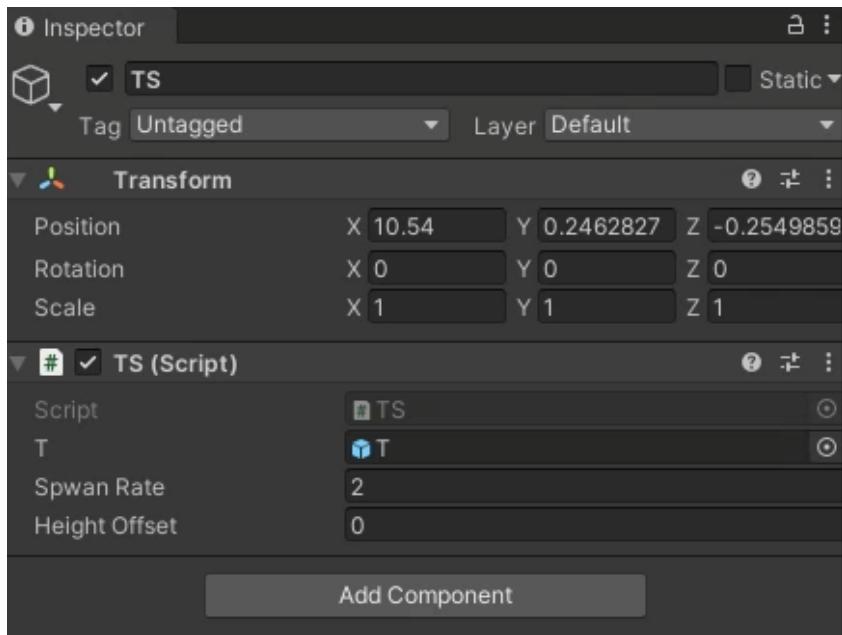
    // Update is called once per frame
    void Update()

    {
        if (timer < spwanRate)
        {
            timer = timer + Time.deltaTime;
        }
        else
        {
            spwanT();
            timer = 0;
        }
    }

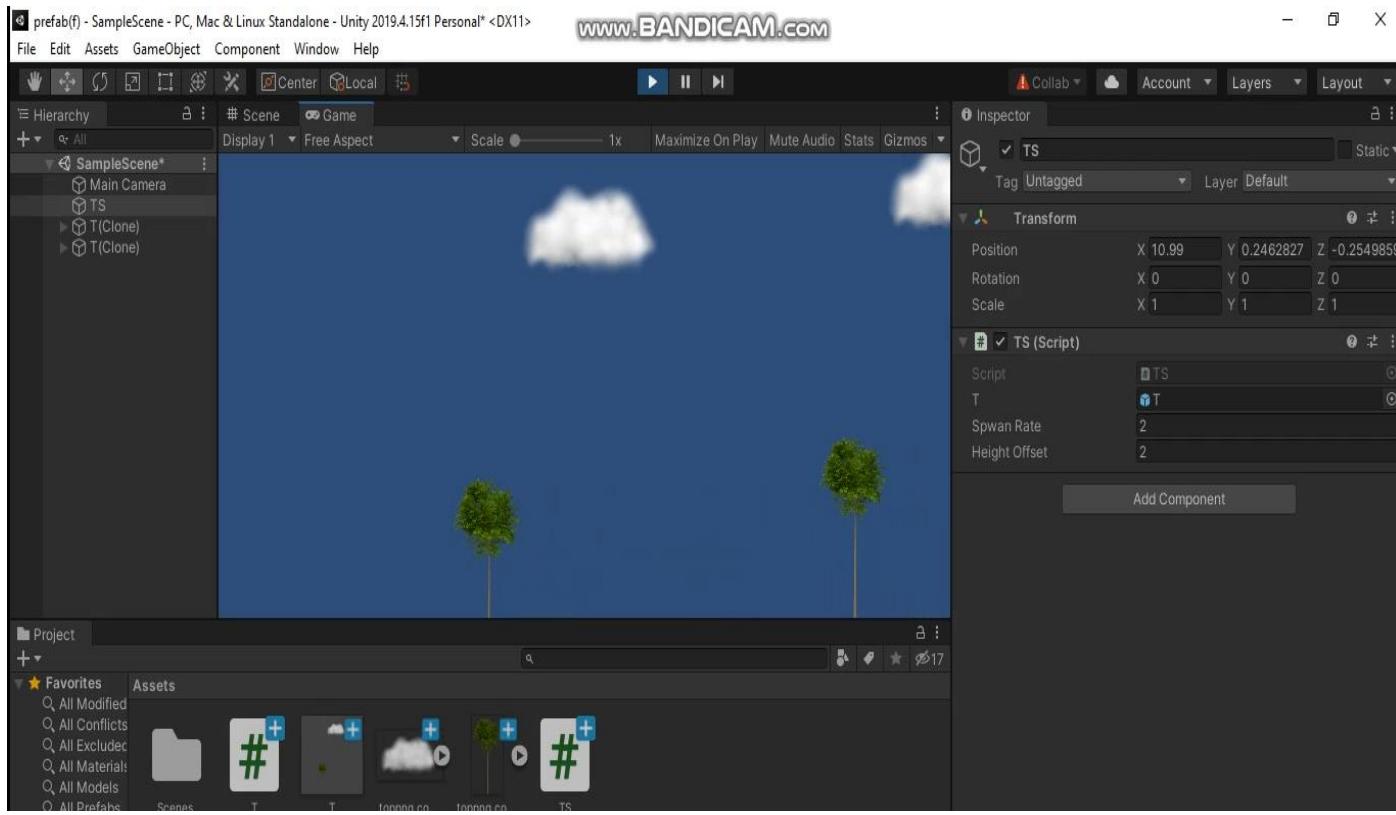
    void spwanT()
    {
        float lowestPoint = transform.position.y - heightOffset;
        float highestPoint = transform.position.y + heightOffset;
        Instantiate(T, new Vector3(transform.position.x, Random.Range(lowestPoint, highestPoint), 0),
        transform.rotation);
    }
}
```

```
 }  
 }
```

Step 8: If incase the pre-fab was not loaded into the new game objects script then you can manually drag and drop it into the new custom field made under the script column



Step 9: Run the Scene



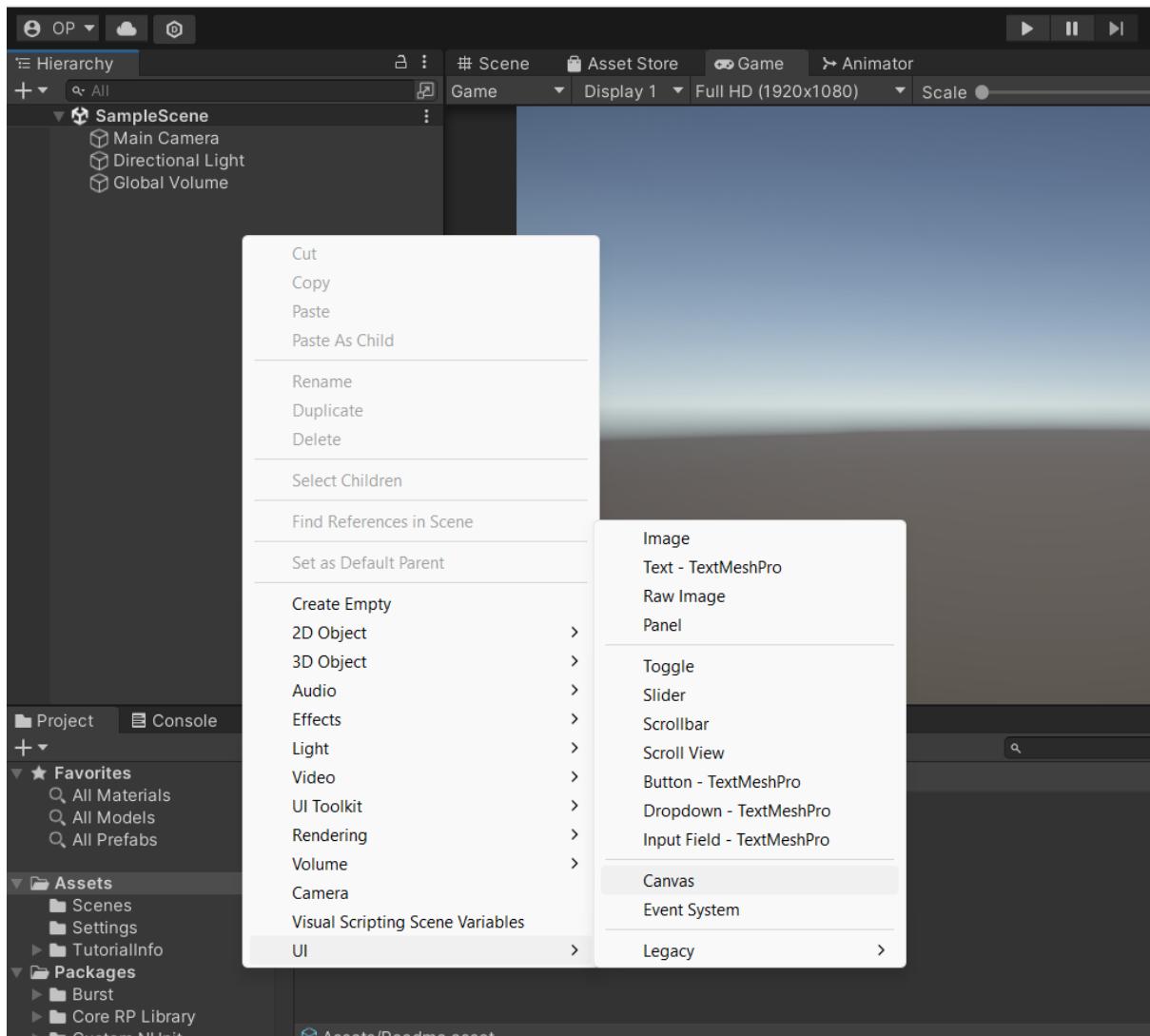
Practical 10: Change of panel on button click.

Open Unity, create new project.

Step 1: Set Up Your UI

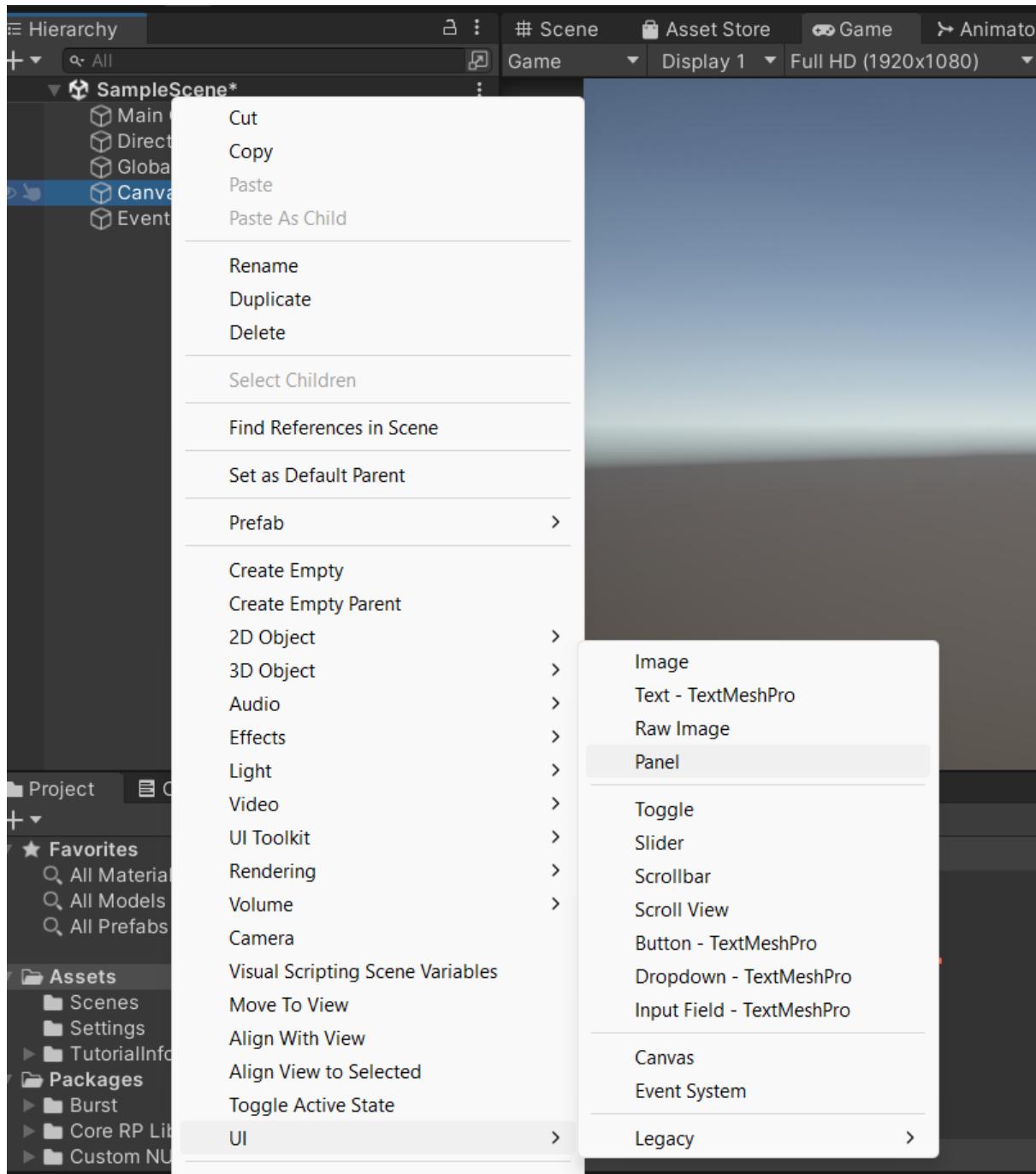
1. Create a Canvas:

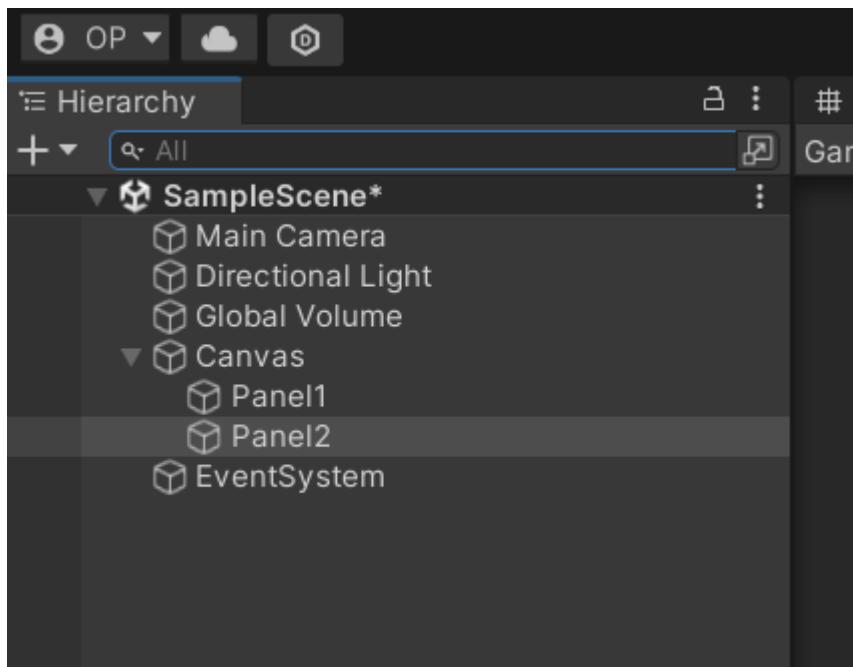
- Right-click in the Hierarchy window.
- Go to **UI > Canvas**.
- This will create a Canvas object in your scene, which will hold all your UI elements.



2. Add Panels:

- Right-click on the Canvas in the Hierarchy.
- Go to **UI > Panel**.
- Rename the first panel to `Panel1` and the second to `Panel2`.
- Adjust their sizes and positions as needed.

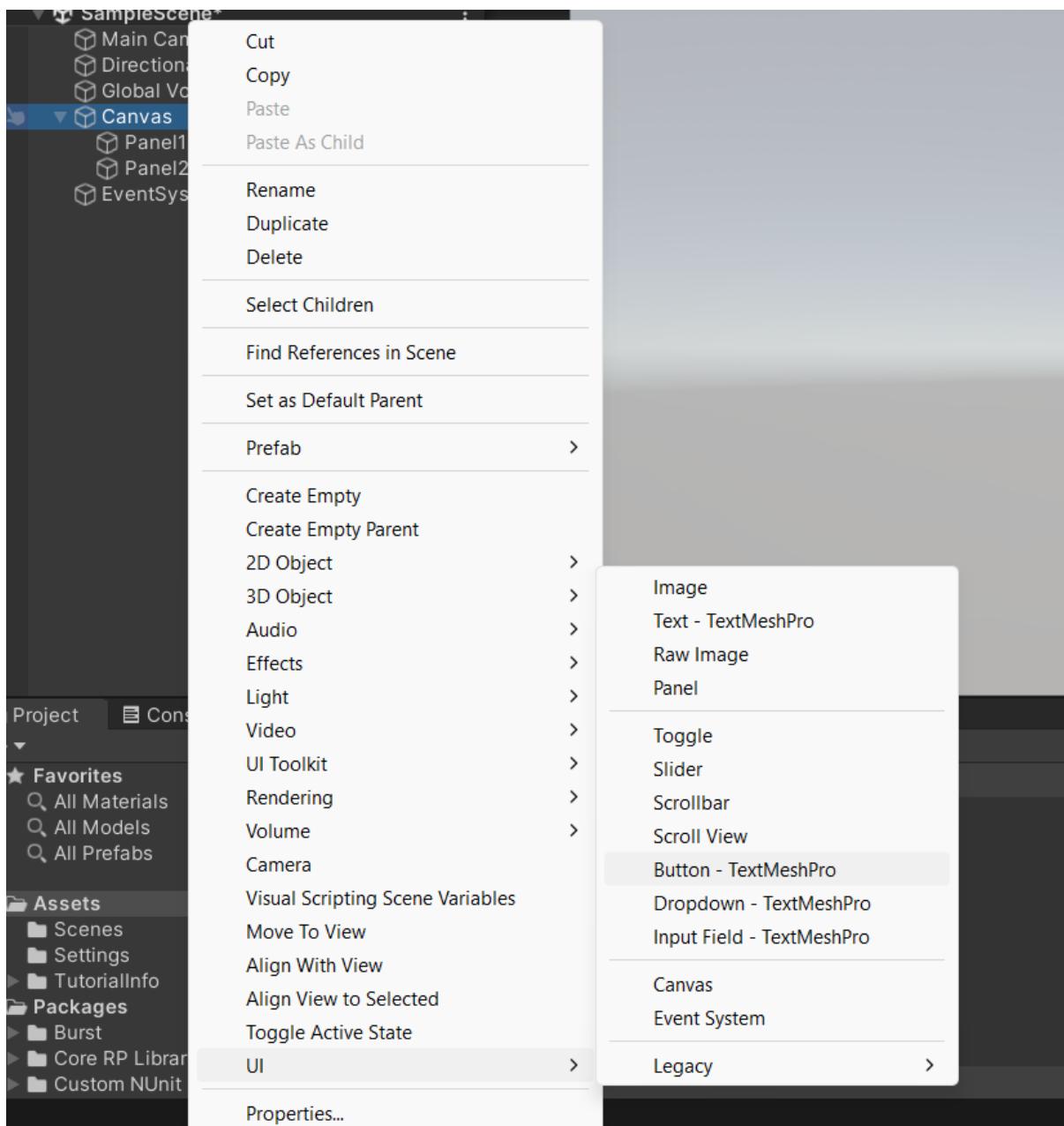




Step 2: Create the Button

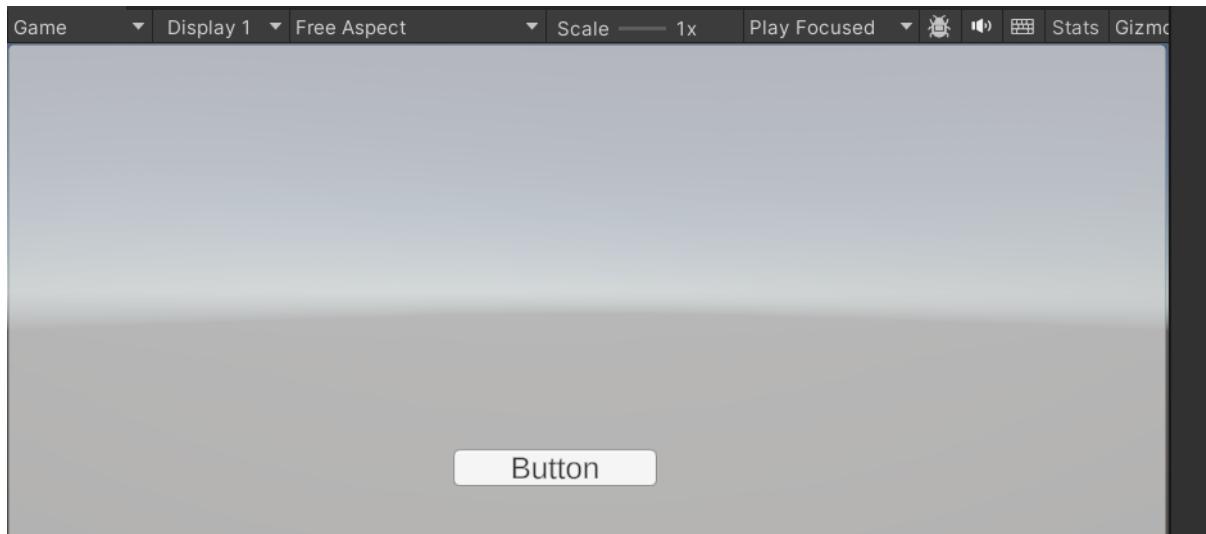
1. Add a Button:

- Right-click on the Canvas.
- Go to **UI > Button**.
- This will create a Button object on your Canvas.



2. Position the Button:

- Select the Button in the Hierarchy.
- Use the Rect Transform tool to position the button where you want it on the Canvas.



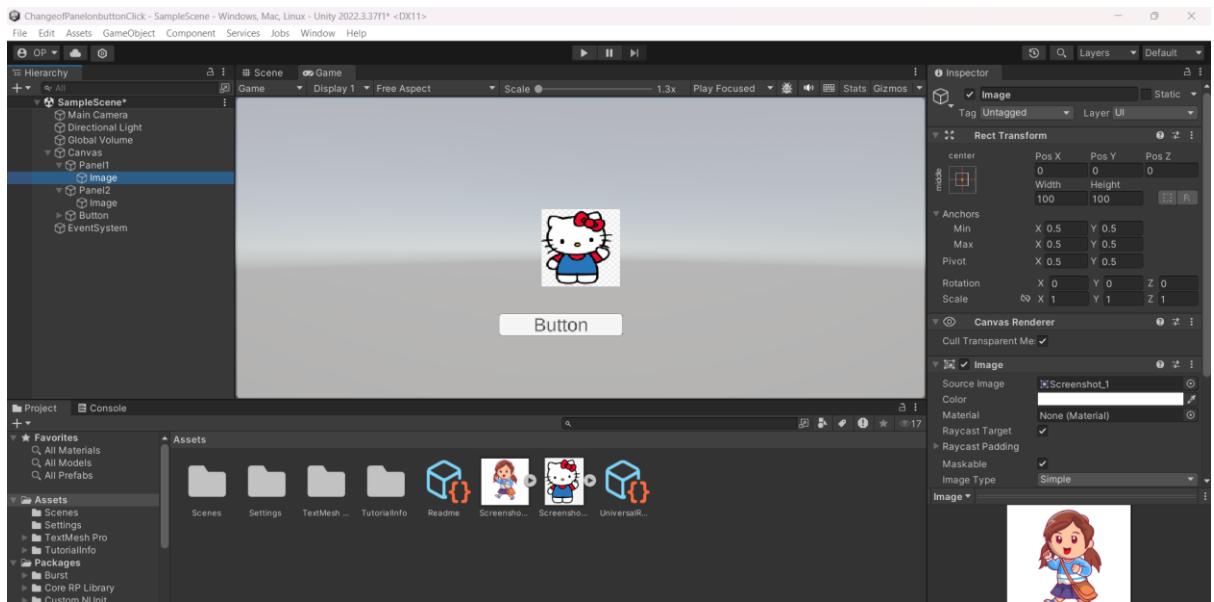
Step 3: Add Images to Panels

1. Import Image Assets:

- Drag and drop your image files into the **Assets** folder in the Project window.

2. Add Image to Panel:

- Select **Panel1** in the Hierarchy.
- Right-click on **Panel1**, go to **UI > Image**.
- In the Inspector, under the **Image (Script)** component, click the small circle next to **Source Image** and select your imported image.
- Adjust the size and position of the image as needed.
- Repeat these steps for **Panel2** if you want an image there as well.



Step 4: Create a Script

1. Create a New Script:

- Right-click in the Project window.
- Go to **Create > C# Script**.

- Name it `PanelSwitcher`.
- 2. Open the Script:**
 - Double-click the script to open it in your code editor.

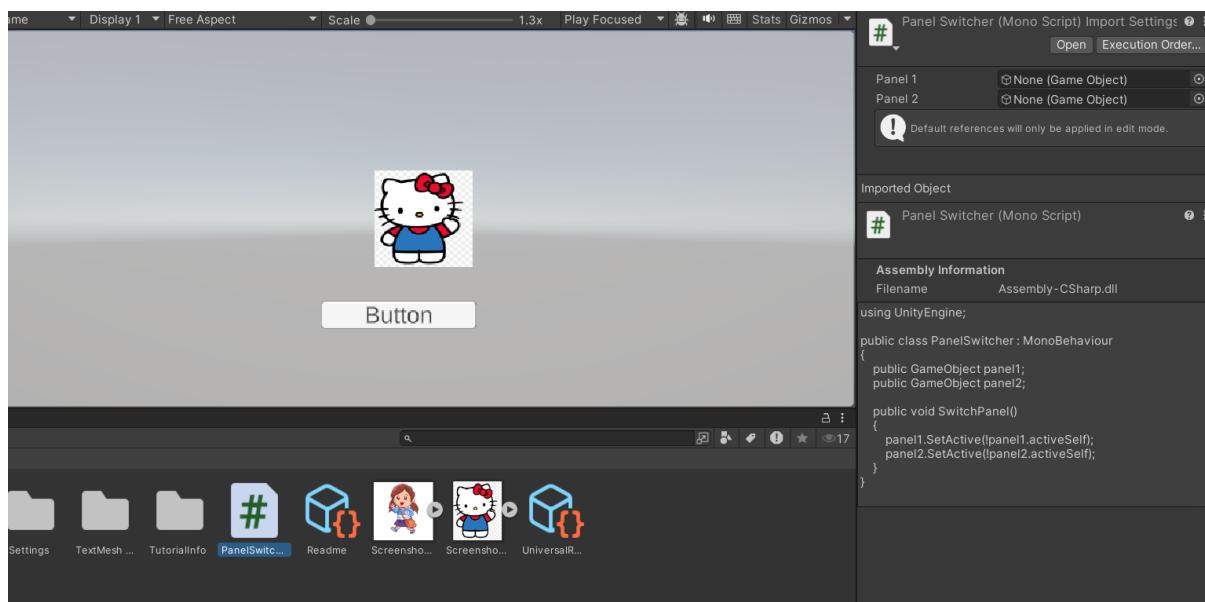
Step 5: Write the Script

Here's a simple script to switch between panels:

```
using UnityEngine;

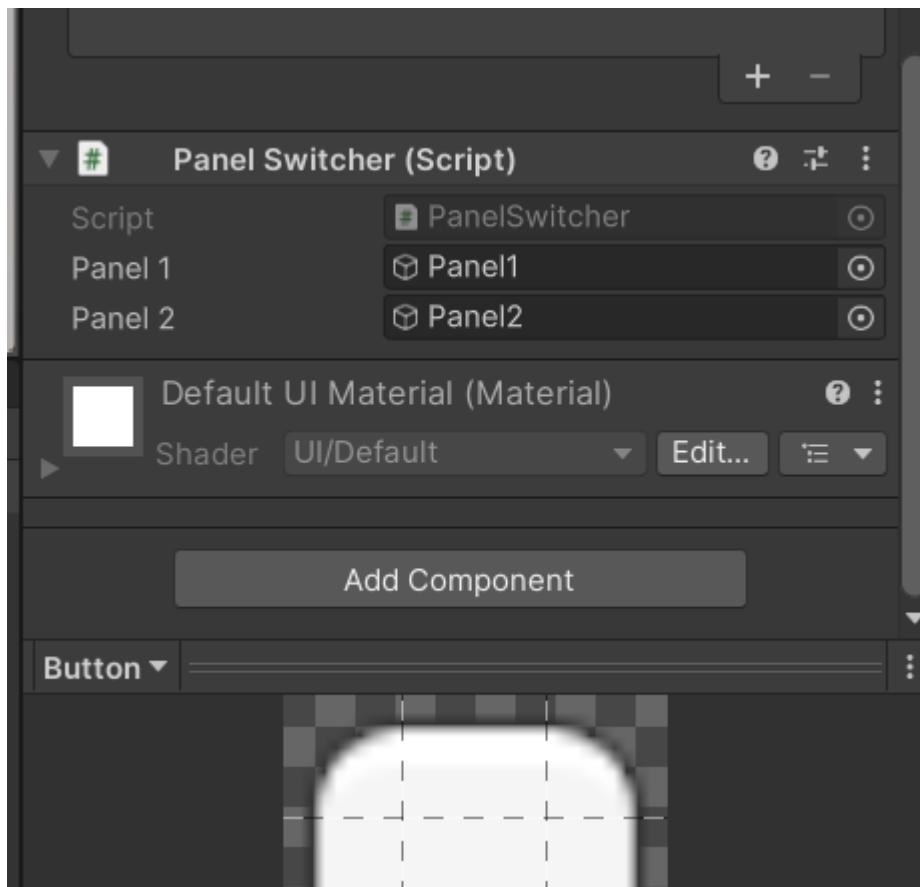
public class PanelSwitcher : MonoBehaviour
{
    public GameObject panel1;
    public GameObject panel2;

    public void SwitchPanel()
    {
        panel1.SetActive(!panel1.activeSelf);
        panel2.SetActive(!panel2.activeSelf);
    }
}
```



Step 6: Attach the Script

- 1. Attach the Script to the Button:**
 - Select the button in the Hierarchy.
 - In the Inspector, click **Add Component** and add the `PanelSwitcher` script.
- 2. Assign Panels:**
 - In the Inspector, drag `Panel1` and `Panel2` from the Hierarchy to the corresponding fields in the `PanelSwitcher` component.



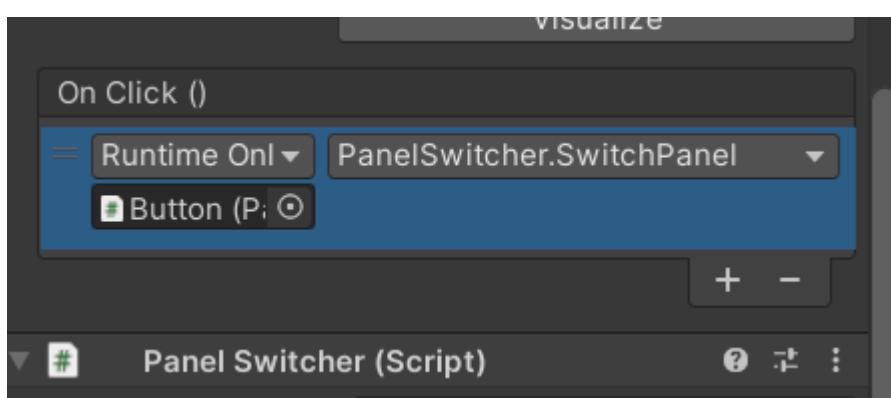
Step 7: Configure the Button

1. Add OnClick Event:

- In the Inspector, scroll down to the **Button (Script)** component.
- Click the + button under **On Click ()**.

2. Assign the Function:

- Drag the button itself into the **Object** field.
- Select **PanelSwitcher > SwitchPanel** from the dropdown menu.



Step 8: Test Your Setup

1. Run the Scene:

- Click the Play button at the top of the Unity editor.

2. Click the Button:

- Clicking the button should now toggle between Panel1 and Panel2.

