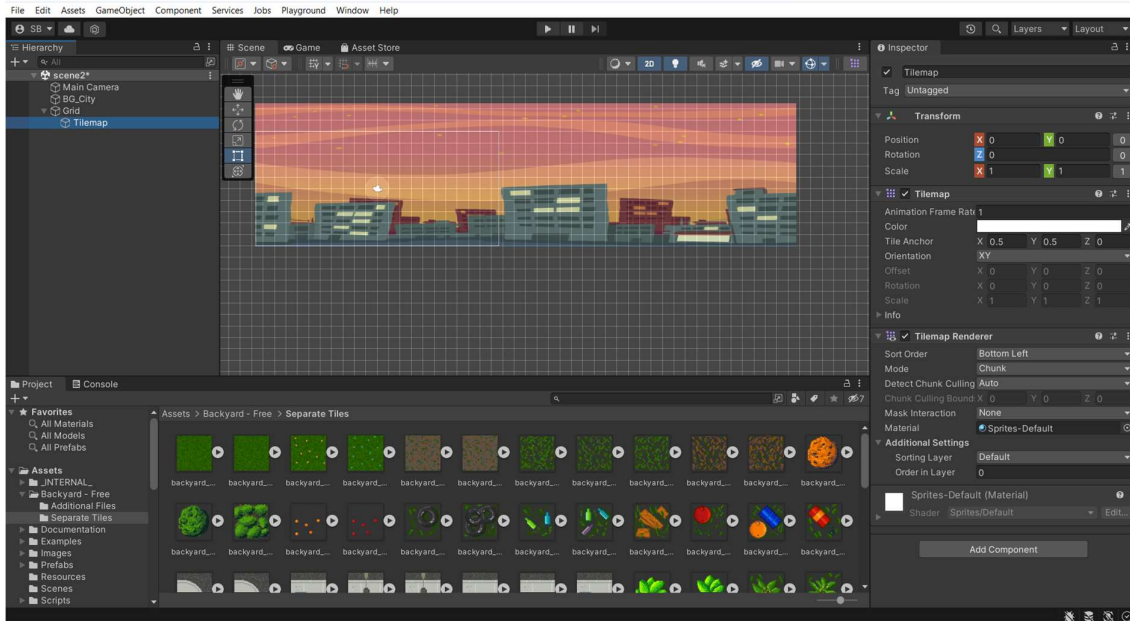# LAB 6
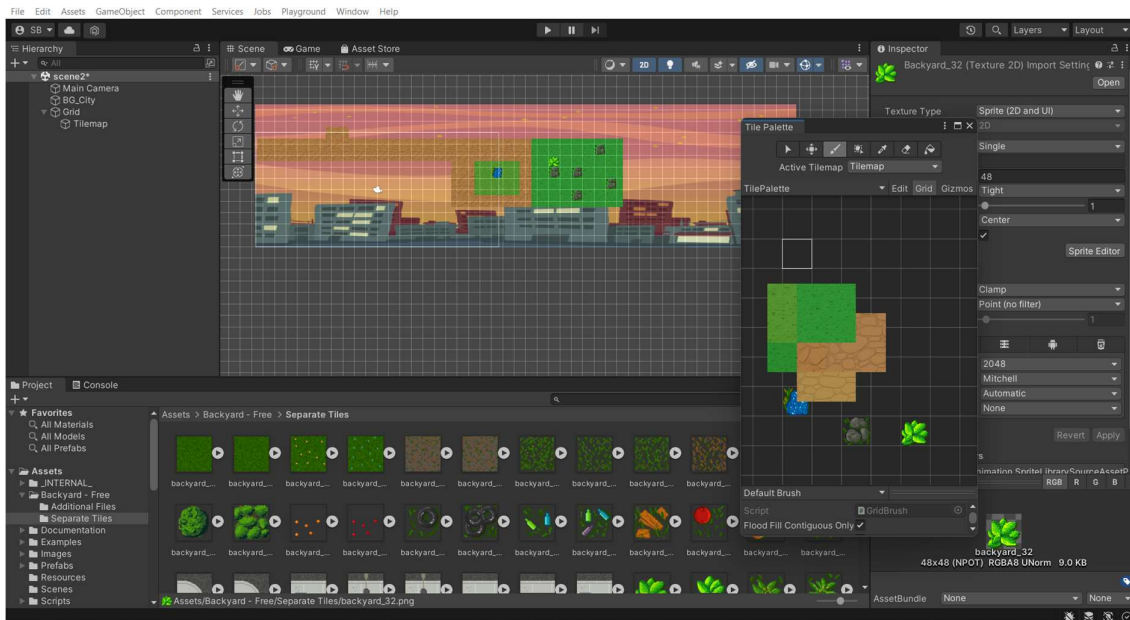
Sudharshanan Balaji

20BAI1242
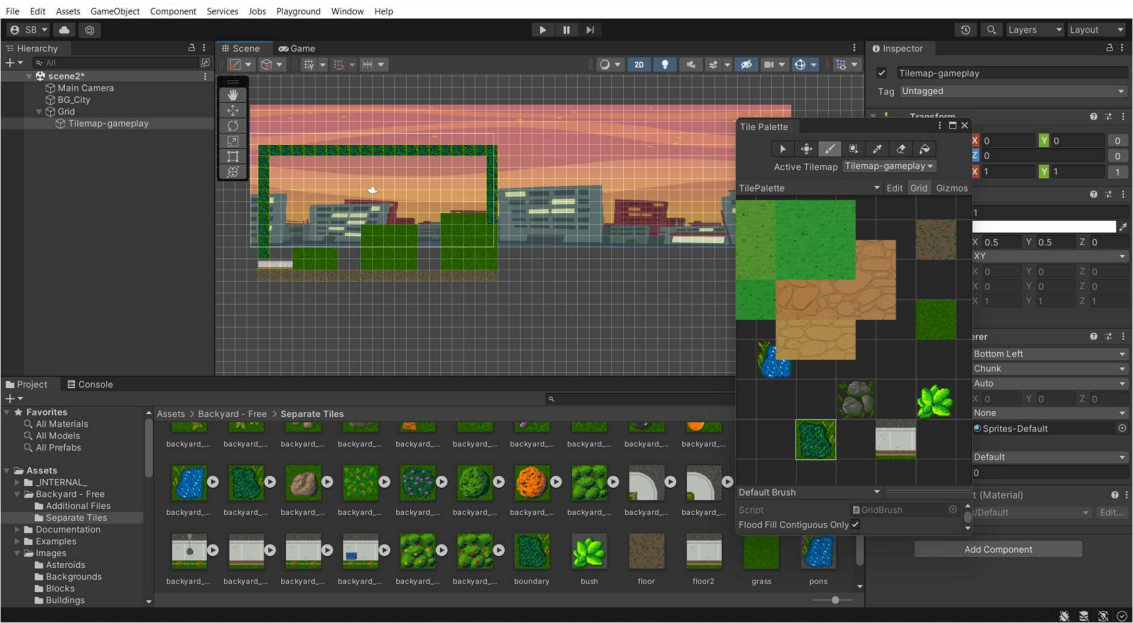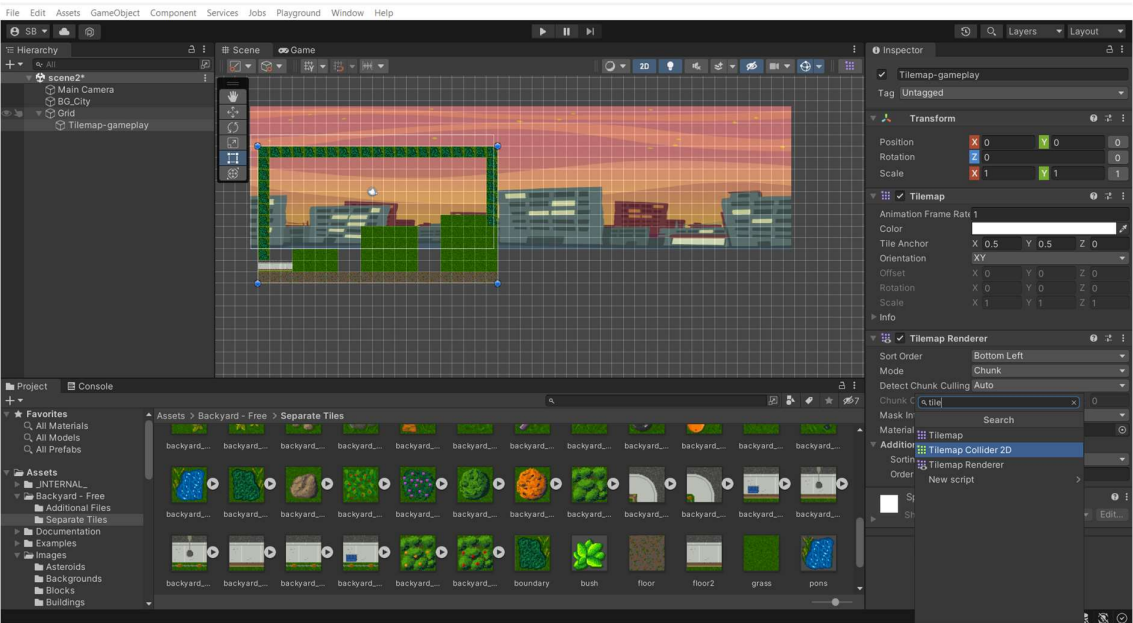
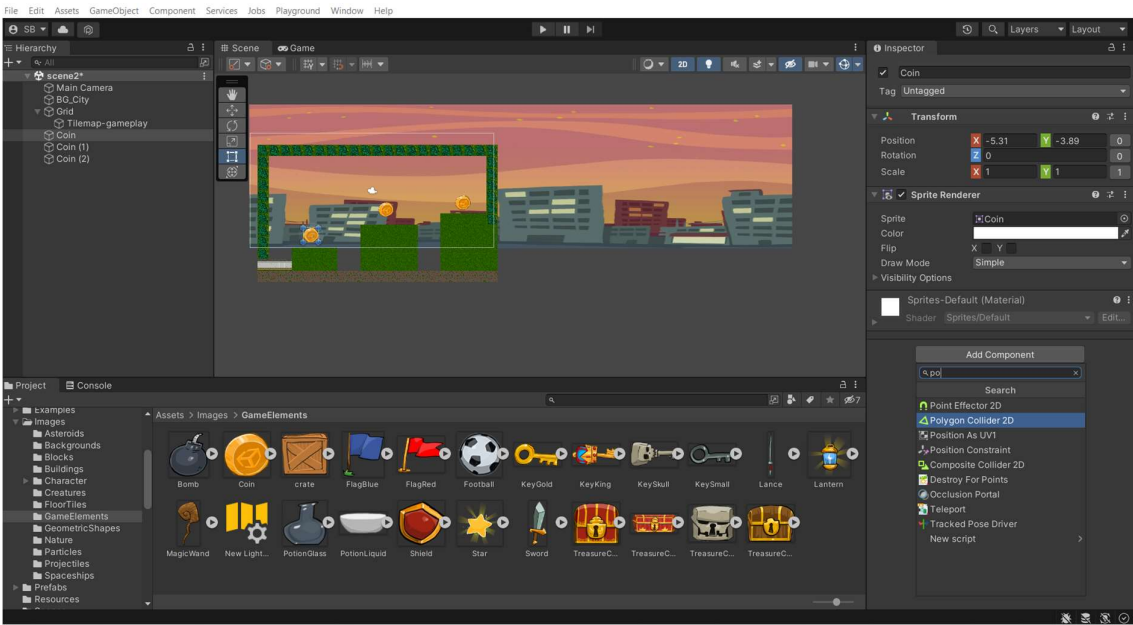Add Tileamap in scene and add a background:

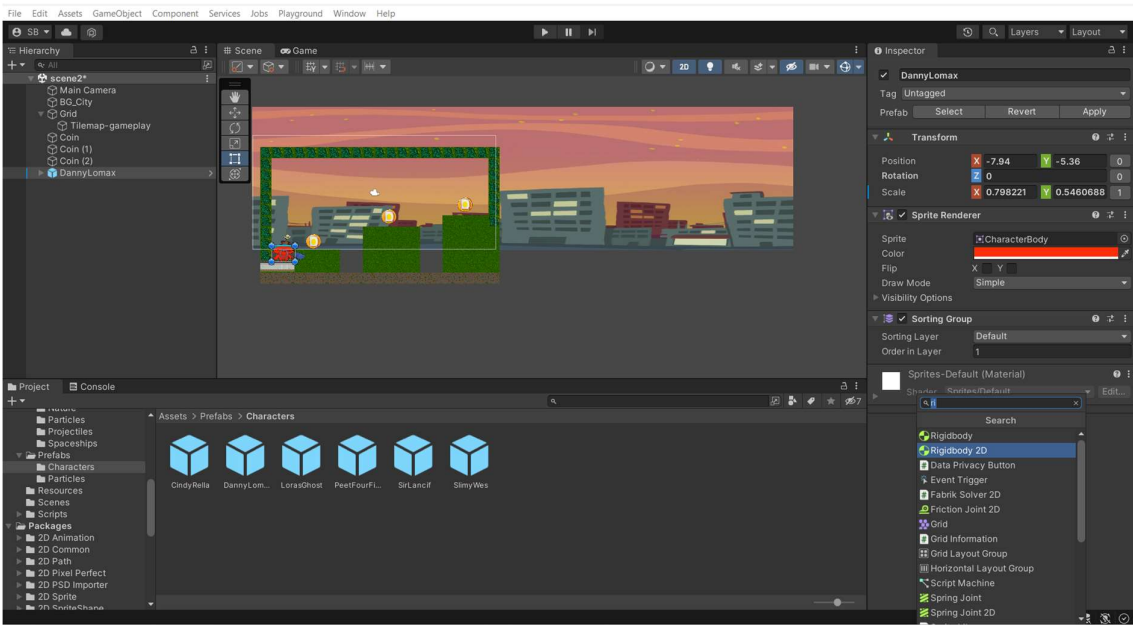Create Tilemap:

Put tiles according to play:
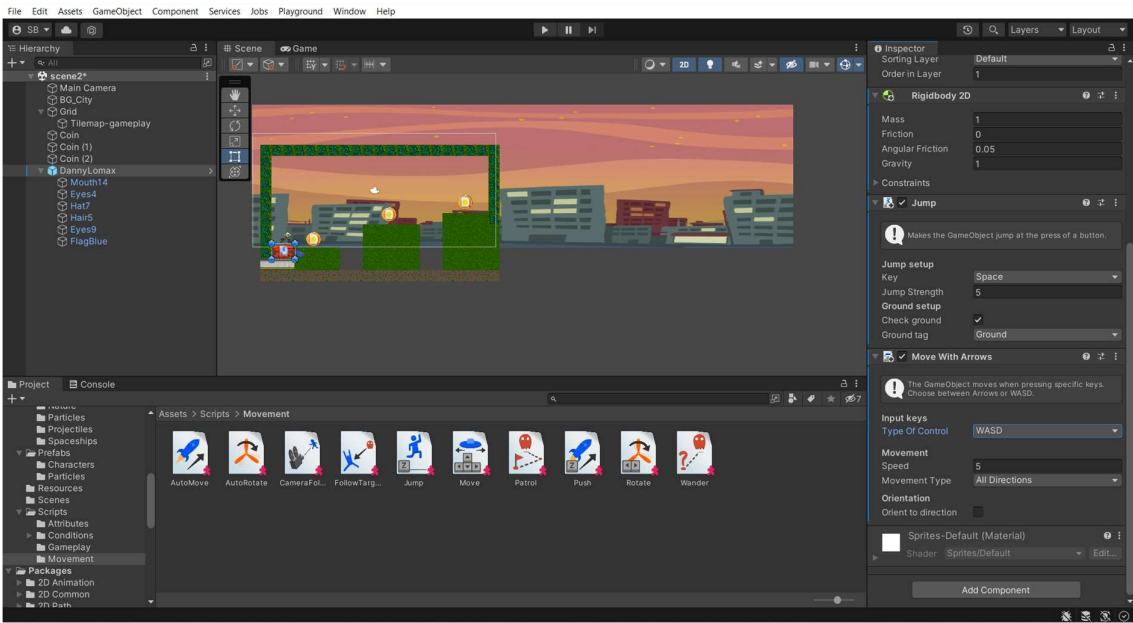

Add Component tilemap collider:

Polygon collider to collectables:

Add charac and rigid body to character:

Add Jump and movement script:

Add slider:
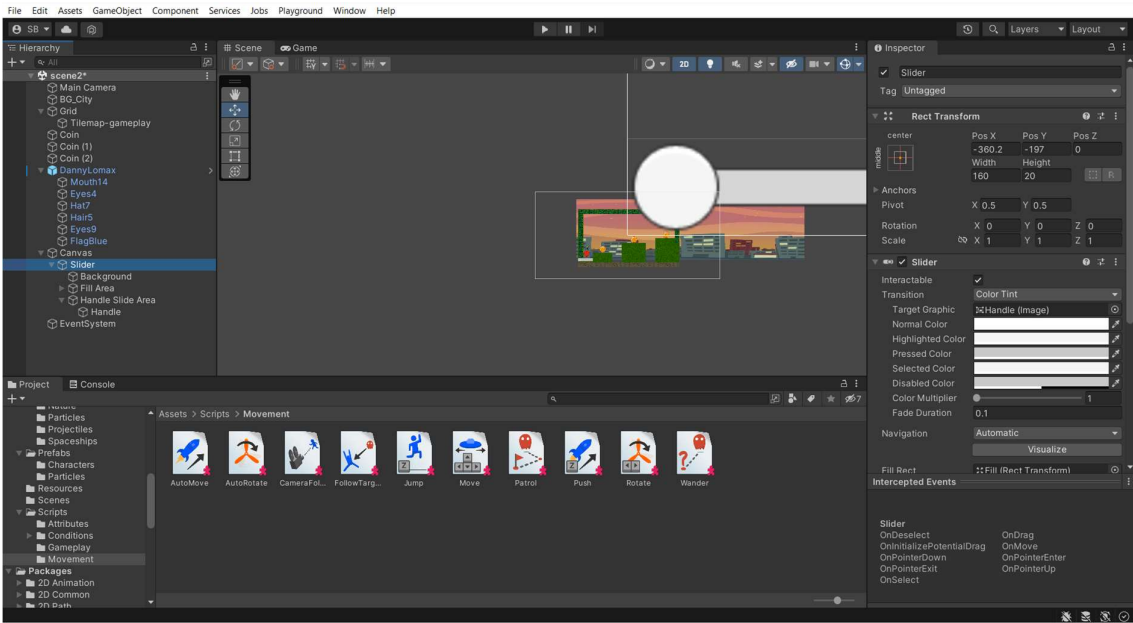
Set slider to main camera
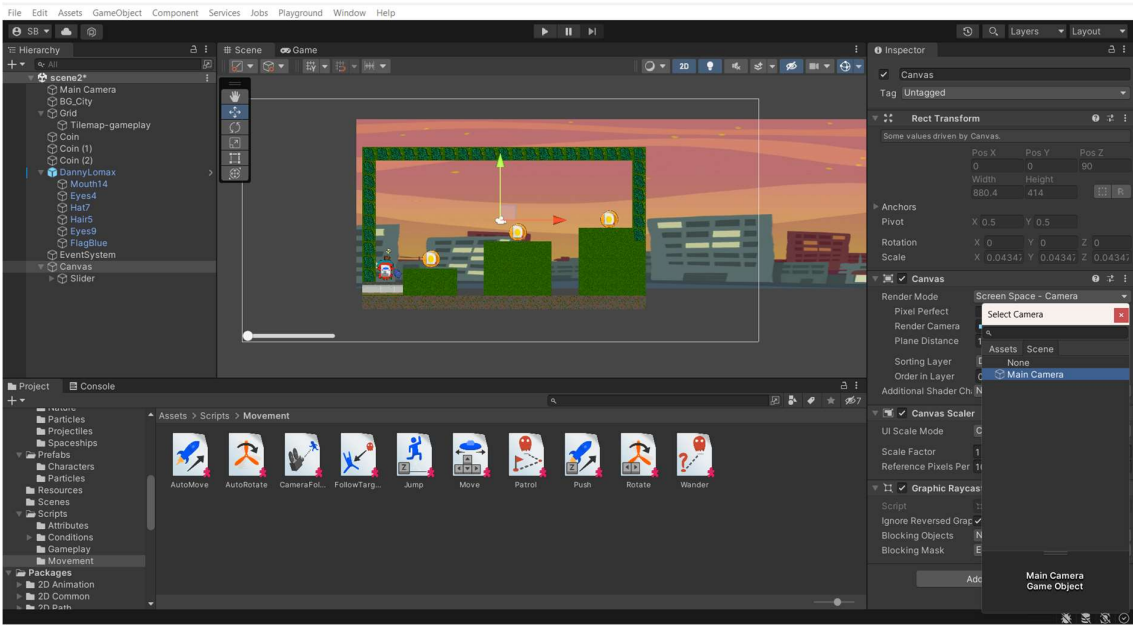
Change the order in layer to 1 in canvas
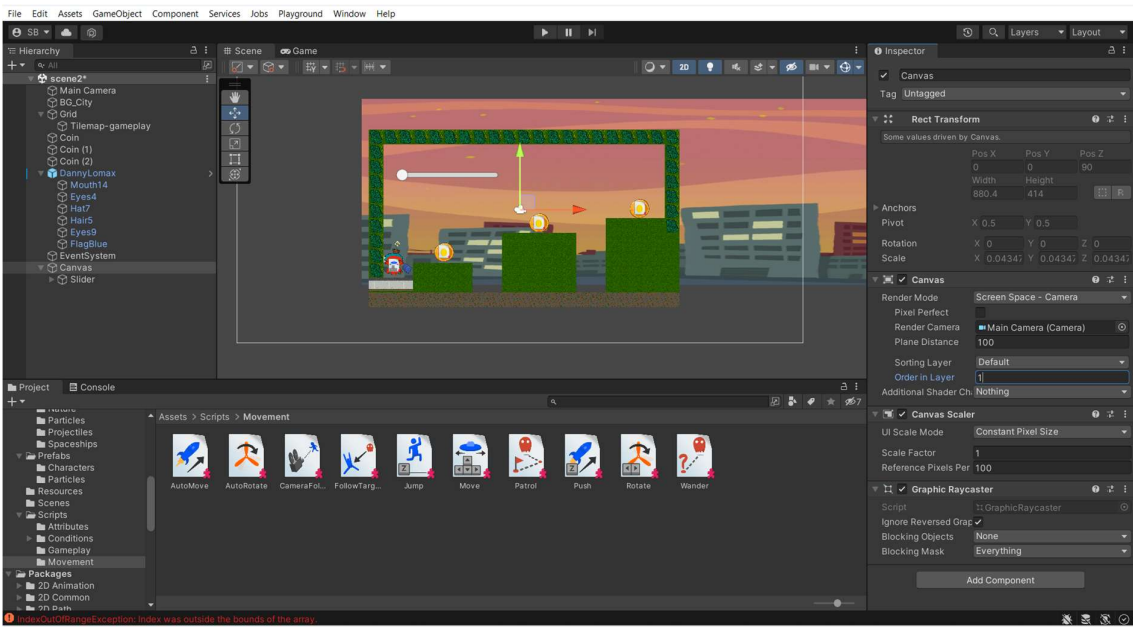
Add a button
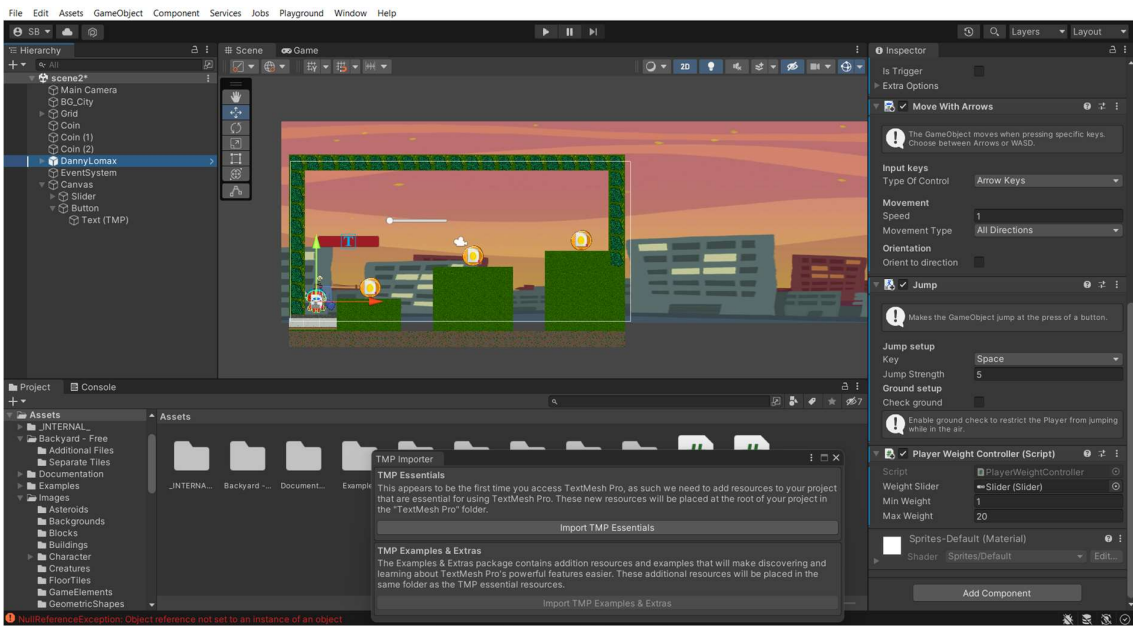
Add prarallax Script to Background and increase speed;
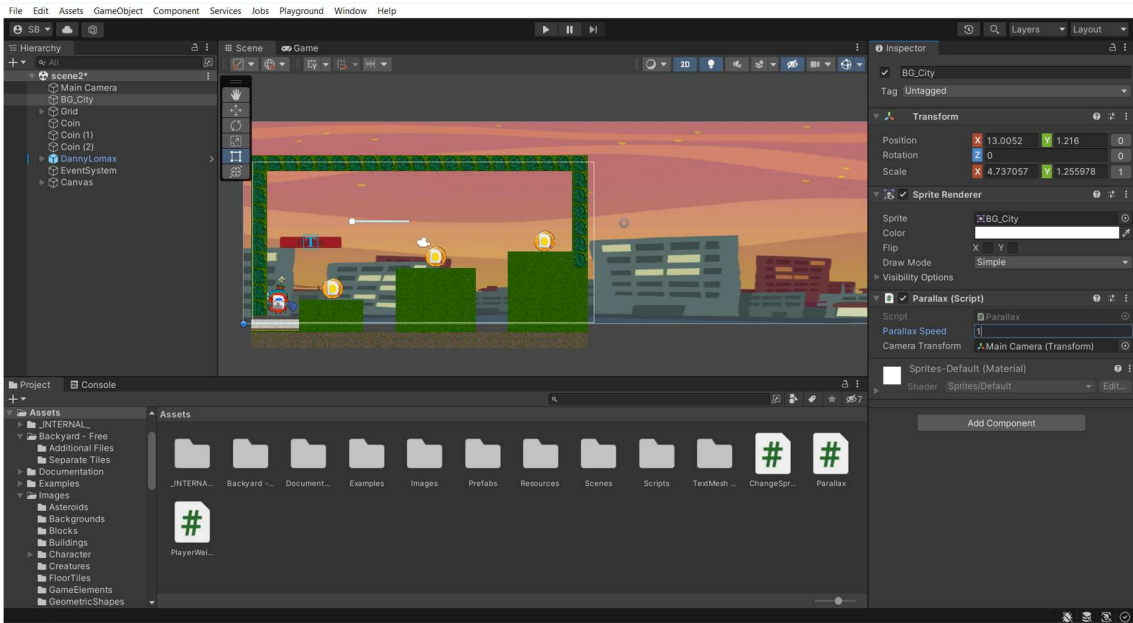
Add User interface prefab

Gameplay Link:

Create a plane and add a material
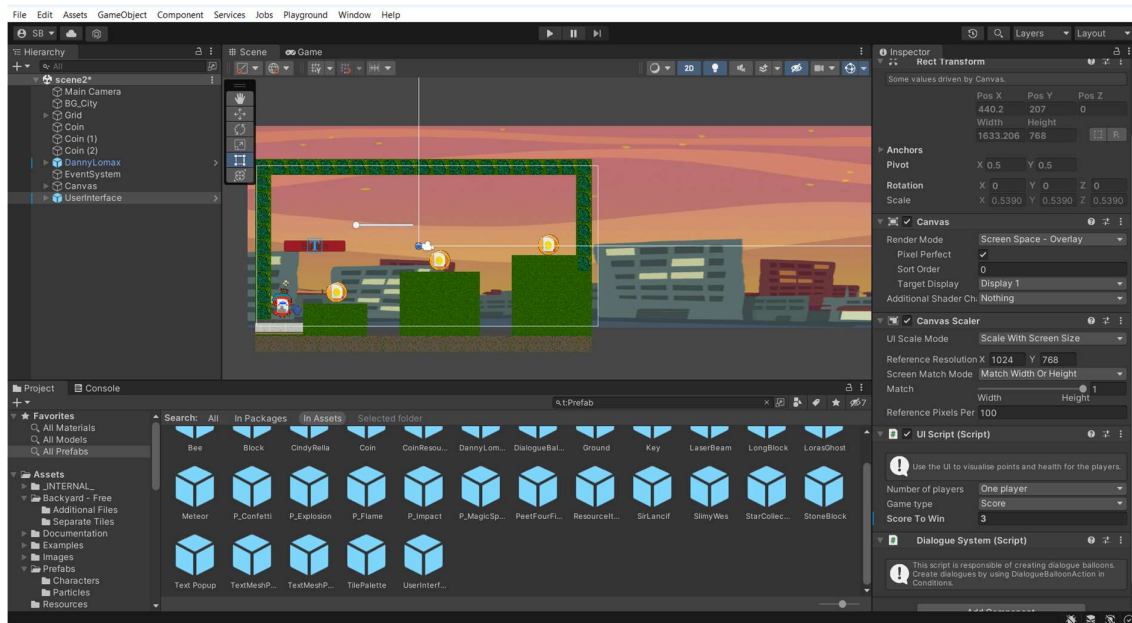
Add Texture:

Game play link:

https://drive.google.com/file/d/163ddY3SFprprymi-vwmiKHlLtAUBmKOp/view?usp=sharing

Scripts used:

Material Scroll

```
using UnityEngine;
public class MaterialScroller : MonoBehaviour {
    public float scrollSpeedX = 0.1f;
    public float scrollSpeedY = 0.1f;
    new private Renderer renderer;
    void Start() {
        renderer = GetComponent<Renderer>();
    }
    void Update() {
        float offsetX = scrollSpeedX * Time.time;
        float offsetY = scrollSpeedY * Time.time;
        renderer.material.mainTextureOffset = new Vector2(offsetX, offsetY);
    }
}
```
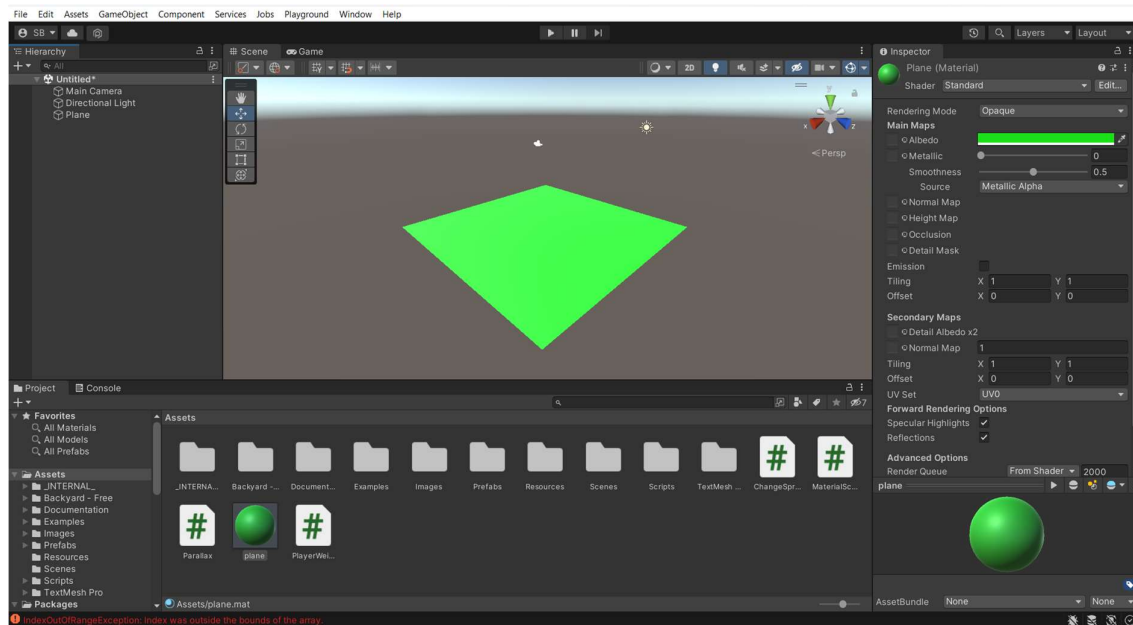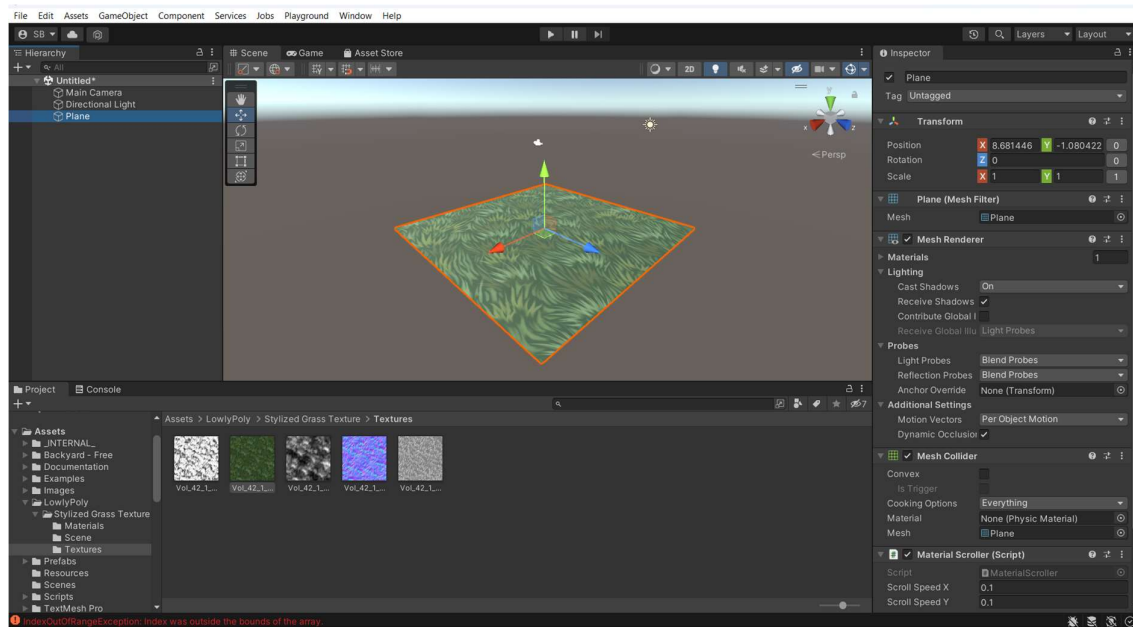
Change color:

```
using UnityEngine;
using UnityEngine.UI;
public class ChangeSpriteColor : MonoBehaviour {
    public SpriteRenderer spriteRenderer;
    public Button colorButton;
    void Start() {
        spriteRenderer = GetComponent<SpriteRenderer>();
        colorButton.onClick.AddListener(xyz);
    }
    void xyz() {
        spriteRenderer.color = Color.blue;
    }
}
```

Parallax:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class Parallax : MonoBehaviour {
    public float parallaxSpeed = 0.02f;
    public Transform cameraTransform;
    private Vector3 lastCameraPosition;
    void Start() {
        lastCameraPosition = cameraTransform.position;
    }
    void FixedUpdate() {
```

```
        float deltaMovement = cameraTransform.position.x -
lastCameraPosition.x;
        Vector3 backgroundPosition = transform.position;
        backgroundPosition.x += deltaMovement * parallaxSpeed;
        transform.position = backgroundPosition;
        if (cameraTransform.position == lastCameraPosition) {
            Vector3 simulatedMovement = new Vector3(parallaxSpeed *
            Time.fixedDeltaTime, 0, 0);
            transform.position -= simulatedMovement;
        }
        lastCameraPosition = cameraTransform.position;
    }
}
```

PlayerWeightControl:

```
using UnityEngine;
using UnityEngine.UI;
public class PlayerWeightController : MonoBehaviour {
    public Slider weightSlider;
    public float minWeight = 1f;
    public float maxWeight = 20f;
    private Rigidbody2D playerRigidbody;
    void Start() {
        playerRigidbody = GetComponent<Rigidbody2D>();
        weightSlider.onValueChanged.AddListener(OnWeightChanged);
    }
    void OnWeightChanged(float value) {
        float weight = Mathf.Lerp(minWeight, maxWeight, value);
        playerRigidbody.mass = weight;
    }
}
```