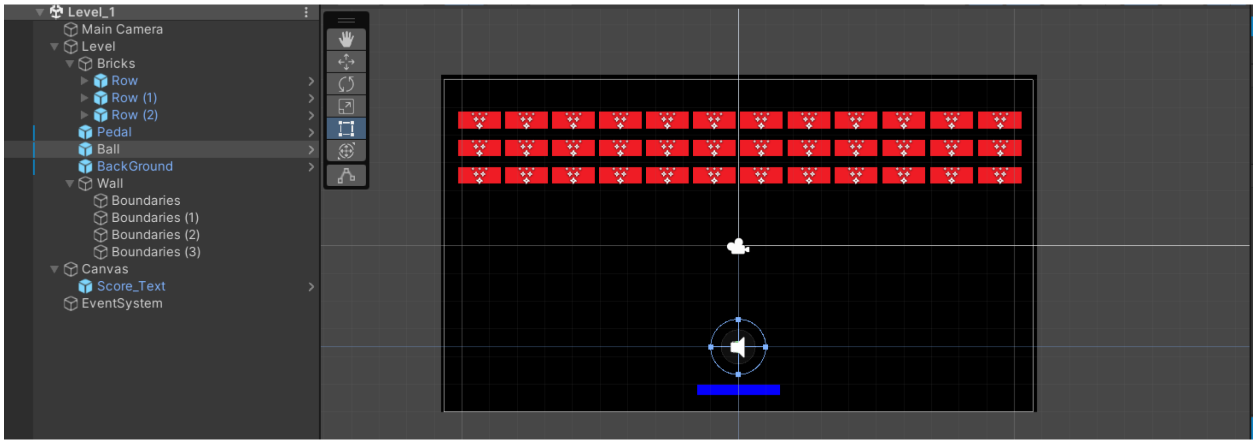
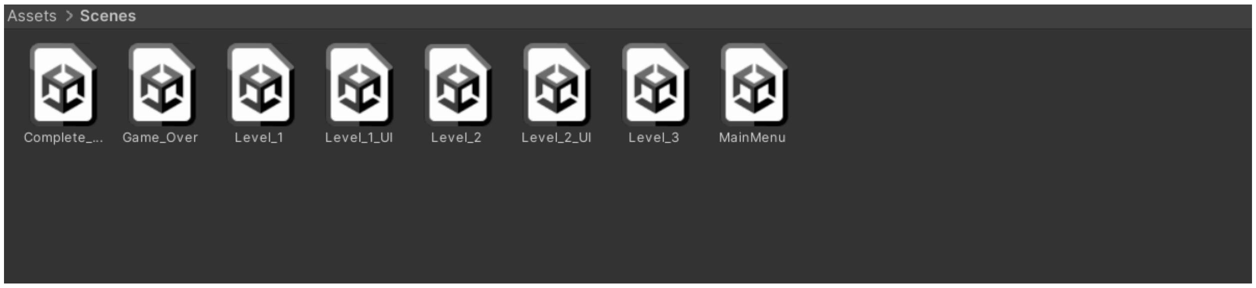
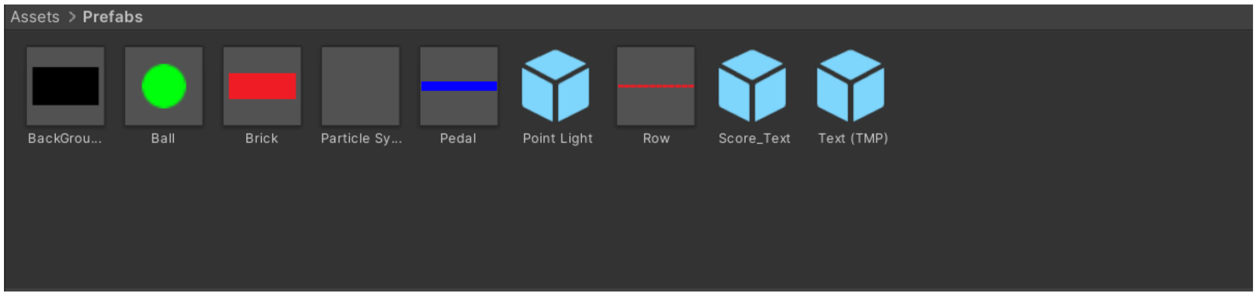
Brick Breaker Game

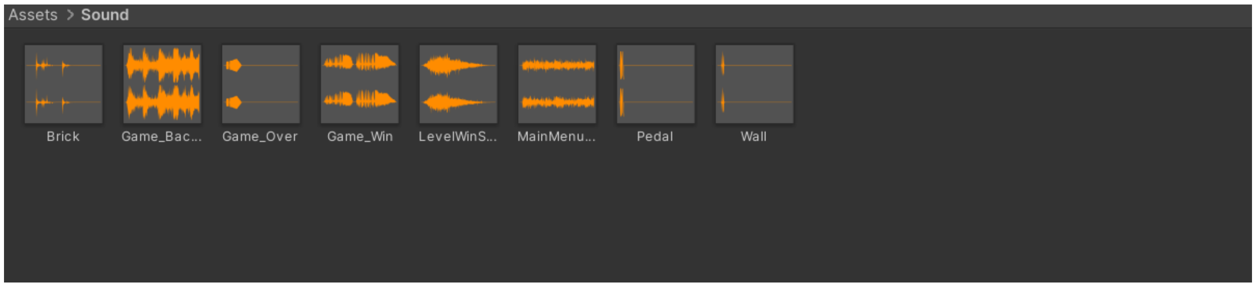
-Level Design

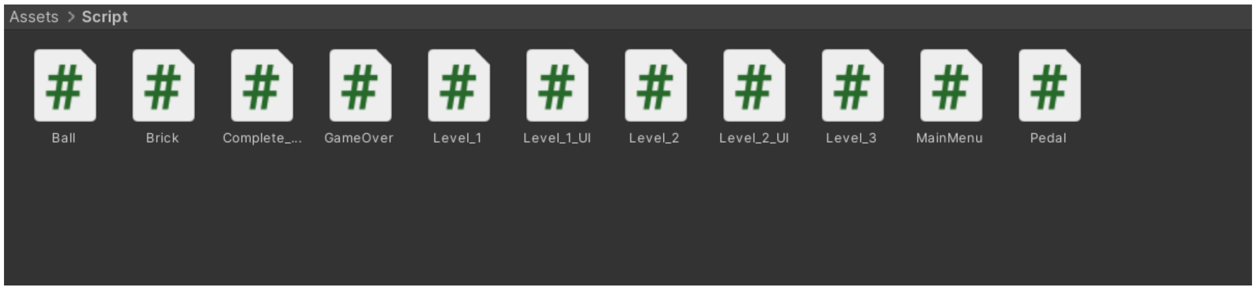
-There are 3 Levels.
-Main Menu,GameOver,LevelComplete Scenes.
-Prefabs,Physics 2D for Bounce,Sound Effects.
-Scripts.











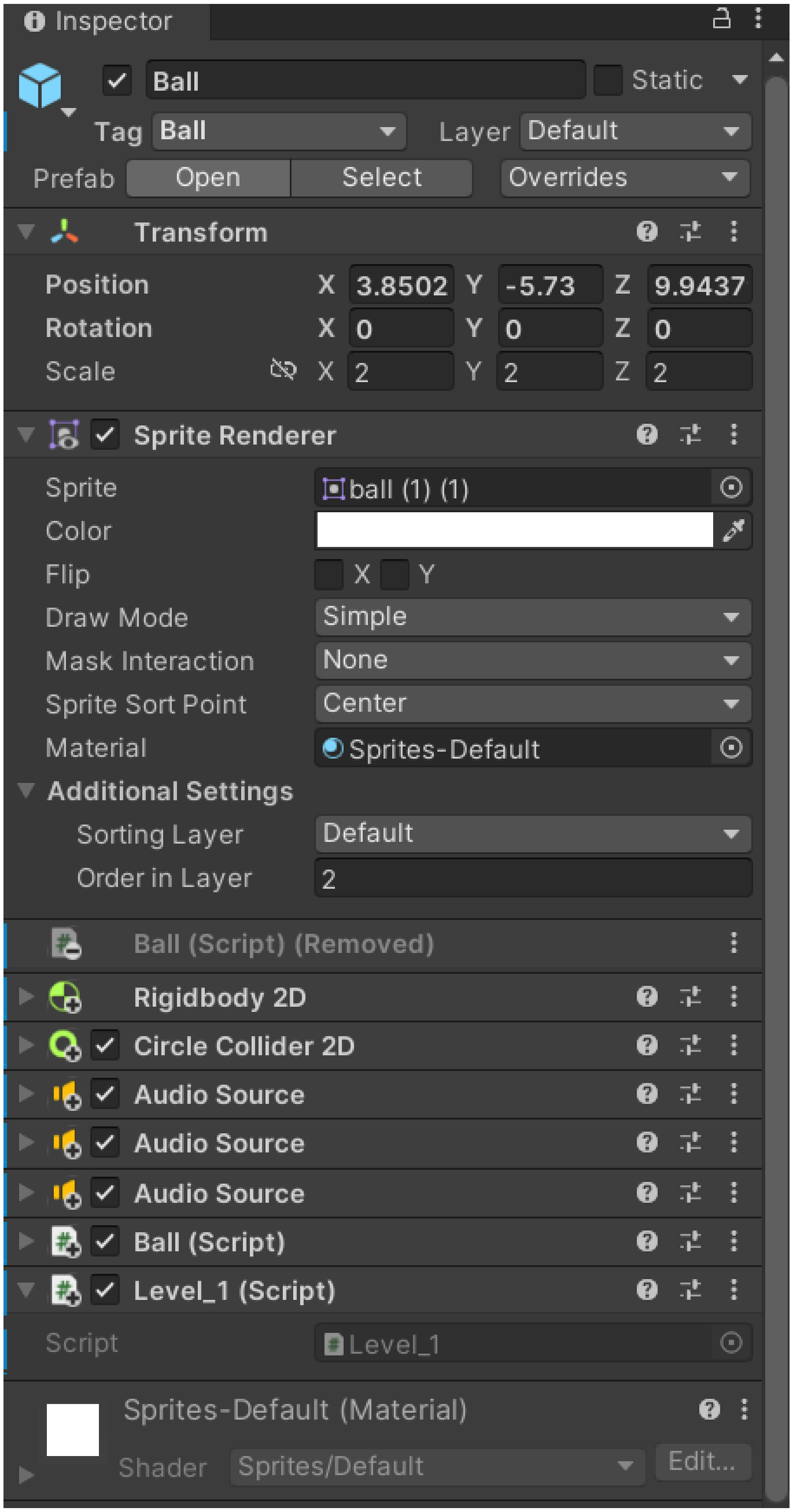
Ball Script

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI; // Importing for UI
using UnityEngine.SceneManagement; // // Importing for SceneManagement

public class Ball : MonoBehaviour
{

    // Initializing and Declaring Fields
    [SerializeField] private AudioSource WallSoundEffect; // Wall Sound
    [SerializeField] private AudioSource PedalSoundEffect; // Pedal Sound
    [SerializeField] private AudioSource BrickSoundEffect; // Brick Sound
    private Rigidbody2D rb; //RigidBody Variable
    private Vector2 force; //Vector2 Variable
    [SerializeField] private float movespeed = 5f;
    public static int Score = 0;
     [SerializeField] private Text ScoreText; //Text Variable that Visible in Unity Engine[SerializeField]
    private float x;
    void Start()
    {
        // When Game Start this code will work for First frame
        rb = GetComponent<Rigidbody2D>();
        Vector2 force = new Vector2();
          force.x = Random.Range(-1f, 1f); //For Random RAnge Max=1f and
Min=-1f
        force.y = -1f;
                rb.AddForce(force.normalized * movespeed); // force with
Normalized Vector - keeping it pointing in the same direction, change its
length to 1.
    }

    void Update()
    {
        // This code will update in each Frame
        x += Time.deltaTime;
        if(x > 3f) // for slowmotion
        {
        rb.velocity = rb.velocity.normalized * movespeed;
        }

    }



    //Function for Brick Collision
    private void Brick(Collision2D Collider)
    {
        if(Collider.gameObject.tag == "Brick")
        {
            Destroy(Collider.gameObject); //Destory Brick
            BrickSoundEffect.Play();
            Score++;
            ScoreText.text = "Score: " + Score;
        }
    }

    //Function for Pedal Collision
    private void Pedal(Collision2D Collider)
    {
        if(Collider.gameObject.tag == "Pedal")
        {
            PedalSoundEffect.Play();
        }
    }

    //Function for Dangerous_Wall Collision
    private void Dangerous_Wall(Collision2D Collider)
    {
        if(Collider.gameObject.tag == "Dangerous_Wall")
        {
            WallSoundEffect.Play();
                SceneManager.LoadScene("Game_Over"); // Loading Game_Over
Scene
        }
    }

    // Function For Collision
    private void OnCollisionEnter2D(Collision2D Collider)
    {
           Dangerous_Wall(Collider); // Dangerous_Wall Collision Function
Call
        Brick(Collider); // Brick Collision Function Call
        Pedal(Collider); // Pedal Collision Function Call
    }
}
```

Pedal Script

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

public class Pedal : MonoBehaviour
{
    // Initializing and Declaring Fields
    private Rigidbody2D rb;
    private float dirX = 0f;
    [SerializeField] private float movespeed = 10f;
    public float Ball_maxBounceAngle = 75f; // Initializing maxBounceAngle

    void Start()
    {
        rb = GetComponent<Rigidbody2D>();
    }

    void FixedUpdate()
    {
            dirX = Input.GetAxisRaw("Horizontal"); // Horizontal/X-axis
Movement <-Left Arrow/Right Arrow->
           rb.velocity = new Vector2(dirX * movespeed,rb.velocity.y); //
Multiply with movespeed
    }

    // Function For Collision with Ball for natural angle
    private void OnCollisionEnter2D(Collision2D ball_Collider)
    {
            Ball ball = ball_Collider.gameObject.GetComponent<Ball>(); //
Fetching ball gameobject in Ball Variable

        if (ball != null) // if Ball Variable is not empty(succeed)
        {
                Vector2 Position = transform.position; // Initializing
Position by current Position of Pedal
                 Vector2 Point = ball_Collider.GetContact(0).point; //
Initializing Point from middle point of Pedal

            float Pedal_offset = Position.x - Point.x; // We will get the
Offset by current Position x coordinate - middle point x coordinate
                                              float  Pedal_maxOffset  =
ball_Collider.otherCollider.bounds.size.x / 2; // We will the max_Offset
of Pedal by full size x coordinates / 2

                     float Ball_Angle = Vector2.SignedAngle(Vector2.up,
ball.GetComponent<Rigidbody2D>().velocity); // current Ball Angle used
SignedAngle(Vector2 from,Vector2 to) for signed Value
```

```
              float Ball_bounceAngle = (Pedal_offset / Pedal_maxOffset) *
Ball_maxBounceAngle; // Ball bounce Angle should be Pedal offset/Pedal
maxOffset *  Ball maxBounceAngle
                        float  Ball_newAngle  =  Mathf.Clamp(Ball_Angle  +
Ball_bounceAngle, -Ball_maxBounceAngle, Ball_maxBounceAngle); // Ball New
Angle  shoulde  use  Math.Clamp(New  Angle  Value,Min  NewAngle  value,Max
NewAngle Value)
             // your angle can't be less than min Angle Value not be more
than maximum Value

                                          Quaternion   Angle_rotation   =
Quaternion.AngleAxis(Ball_newAngle,      Vector3.forward);//In      Unity
Quaternions are used to represent rotations.
             //They are generalization of two-dimensional complex numbers
to three dimensions. AngleAxis(Angle Value,Vector3.direction).
             ball.GetComponent<Rigidbody2D>().velocity = Angle_rotation *
Vector2.up * ball.GetComponent<Rigidbody2D>().velocity.magnitude;
             // for velocity of angle = angle_rotation * direction * speed
of ball
        }
   }
}
```


Inspector

✓ Ball Static

Tag Ball Layer Default

Prefab Open Select Overrides

Transform

Position	X	3.8502	Y	-5.73	Z	9.9437
Rotation	X	0	Y	0	Z	0
Scale	X	2	Y	2	Z	2

✓ **Sprite Renderer**

Sprite	ball (1) (1)
Color	
Flip	X Y
Draw Mode	Simple
Mask Interaction	None
Sprite Sort Point	Center
Material	Sprites-Default

Additional Settings

Sorting Layer	Default
Order in Layer	2

Ball (Script) (Removed)

Rigidbody 2D

✓ **Circle Collider 2D**

✓ **Audio Source**

✓ **Audio Source**

✓ **Audio Source**

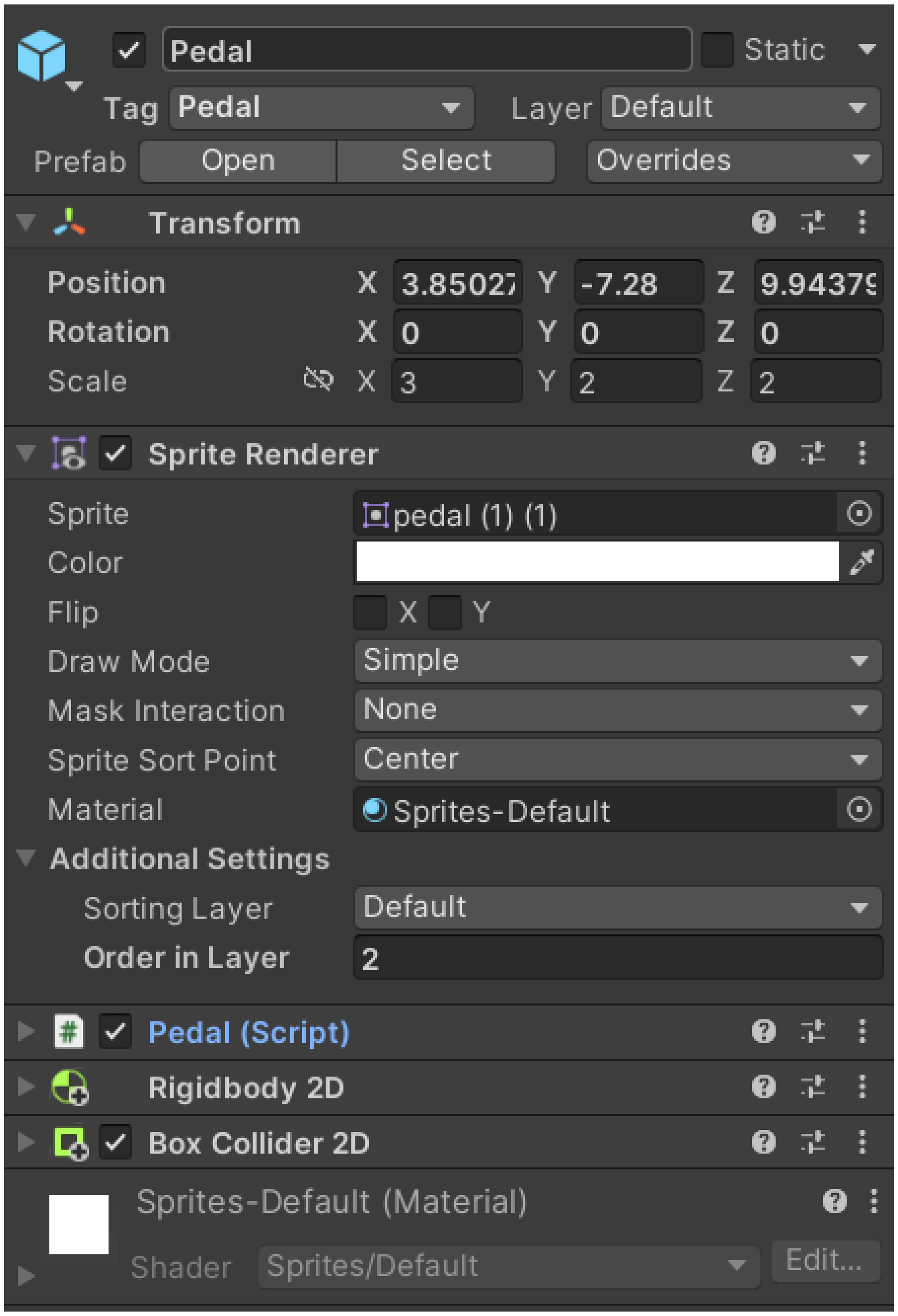
✓ **Ball (Script)**

✓ **Level_1 (Script)**

Script Level_1

Sprites-Default (Material)

Shader Sprites/Default Edit...



Pedal ✓ Static
Tag Pedal Layer Default
Prefab Open Select Overrides

Transform

Position	X	3.85027	Y	-7.28	Z	9.94379
Rotation	X	0	Y	0	Z	0
Scale	X	3	Y	2	Z	2

Sprite Renderer ✓

Sprite	pedal (1) (1)
Color	⬜
Flip	X Y
Draw Mode	Simple
Mask Interaction	None
Sprite Sort Point	Center
Material	Sprites-Default

Additional Settings

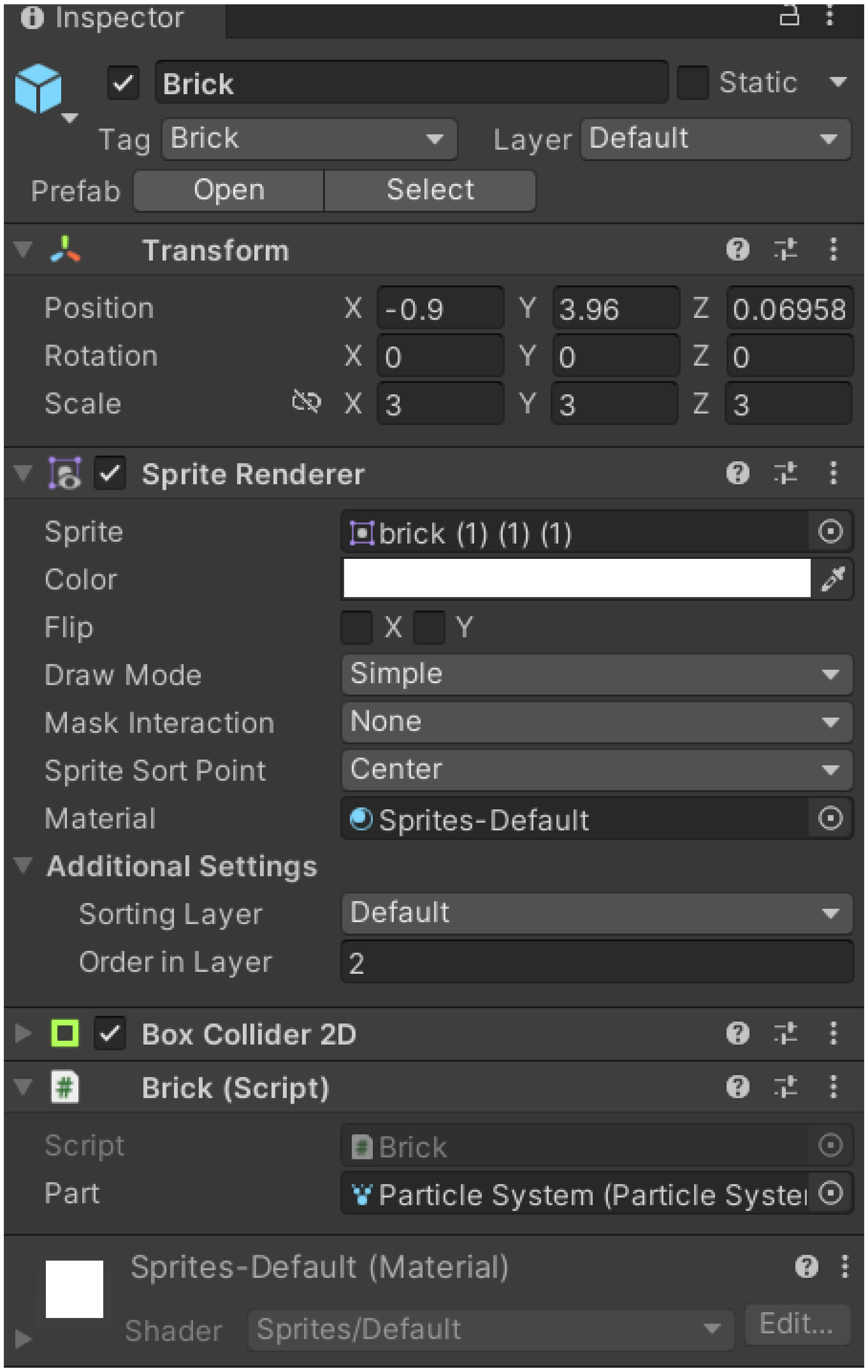
Sorting Layer	Default
Order in Layer	2

Pedal (Script) ✓

Rigidbody 2D

Box Collider 2D ✓

Sprites-Default (Material)
Shader Sprites/Default Edit...


Inspector

✓ Brick Static
Tag Brick Layer Default
Prefab Open Select

Transform
Position X -0.9 Y 3.96 Z 0.06958
Rotation X 0 Y 0 Z 0
Scale X 3 Y 3 Z 3

✓ Sprite Renderer
Sprite: brick (1) (1) (1)
Color
Flip X Y
Draw Mode: Simple
Mask Interaction: None
Sprite Sort Point: Center
Material: Sprites-Default
Additional Settings
Sorting Layer: Default
Order in Layer: 2

✓ Box Collider 2D

Brick (Script)
Script: Brick
Part: Particle System (Particle Syste

Sprites-Default (Material)
Shader Sprites/Default Edit...