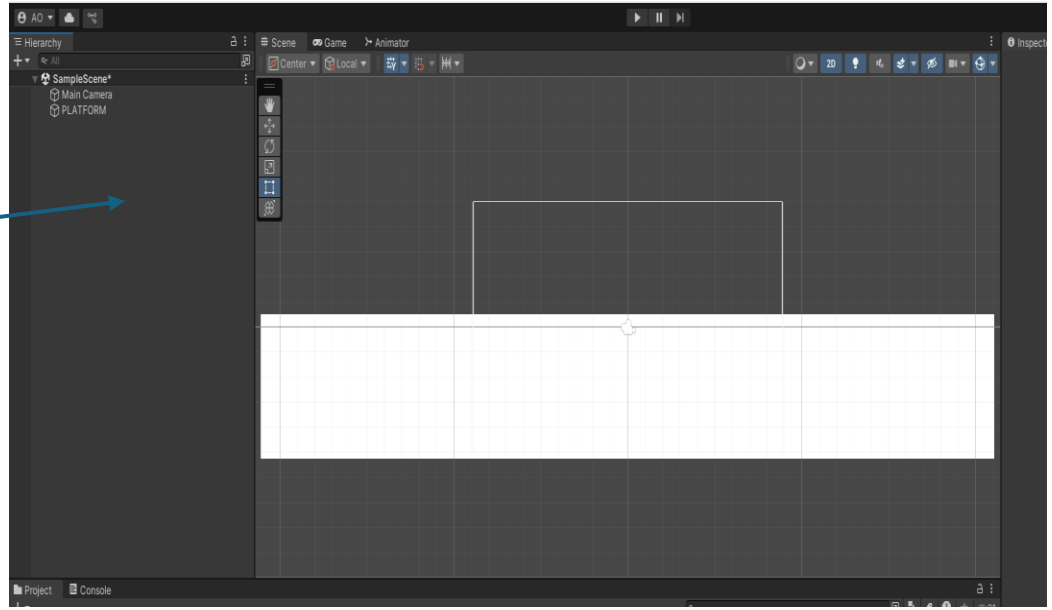
2D MOVEMENT

Setting up the pieces

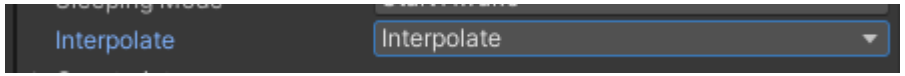


First you're going to right click in the hierarchy and create a square sprite. This is going to be the platform.

You're then going to extend the square as much as you wish since it will be acting as your floor.

You'll then create another square and name it Player.

Once you've done that you're then going to add A [Box Collider 2D] and a [Rigidbody 2D] to the player

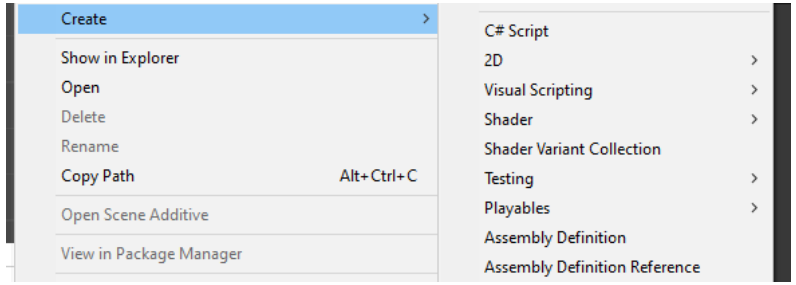


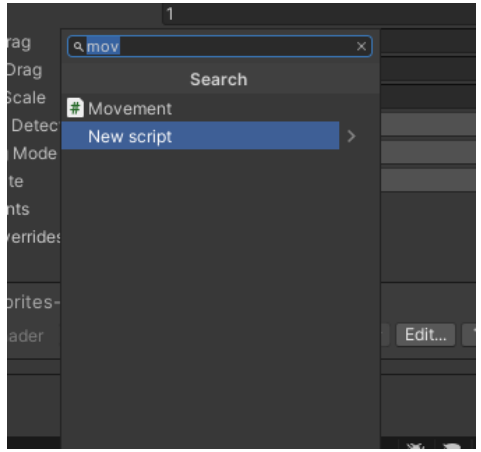
Then you're going to scroll to Rigidbody 2D and change the Collision Detection to continuous and then change Interpolate to interpolate.

Changing these two settings will ensure the movement runs smoothly

The next step is to now create a script called [Movement] as the name states this script will be what enables you to move left and right.

Script for movement





- To add the script you're going to right click in the project area click create -> c# script
- Then your player sprite and click add component then you're going to type the script (Movement) and add it in
- Then you're going to open the script.

Script for Movement PT2

- This is the SCRIPT you'll be typing out: it will be explained step by step
- The public and private variables: public means that that the variable which is speed, will be available to edit in the unity editor. You'll be able to adjust the speed of the player
- The private variable which is Move is pretty much the variable that allows the player to move horizontally which will be referenced later down in the code.
- Rb; will allow the rigidbody to be accessed but not able to be edited as it is set as a private variable. This will apply a sense of realistic physics to the player
- Move = Input.GetAxis("Horizontal) applies horizontal movement for the player automatically. It will auto assign the horizontal movement to the keys A and D, (left and right).
- Move * speed: this code means that the horizontal movement will then be applied to speed. This tie the movement and speed together, thus allowing you to manipulate the speed in unity editor
- Now we will combine move *speed with the rest of the code - rb.velocity = new Vector2(Move * speed, rb.velocity.y); This piece of code affects the velocity of Rigidbody2D as well as the horizontal velocity. This makes sure the player still moves left and right but is still affected by forces of Rigidbody2D.

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
    Unity Script (1 asset reference) | 0 references
5   public class Movement : MonoBehaviour
6   {
7       public float speed;
8       private float Move;
9
10
11      private Rigidbody2D rb;
12      // Start is called before the first frame update
            Unity Message | 0 references
13      void Start()
14      {
15          rb= GetComponent<Rigidbody2D>();
16      }
17
18      // Update is called once per frame
            Unity Message | 0 references
19      void Update()
20      {
21          Move = Input.GetAxis("Horizontal");
22          rb.velocity = new Vector2(Move * speed, rb.velocity.y);
23      }
24  }
25
```

REFERENCE

- https://github.com/DanRex9/2D-Movement-Tutorial