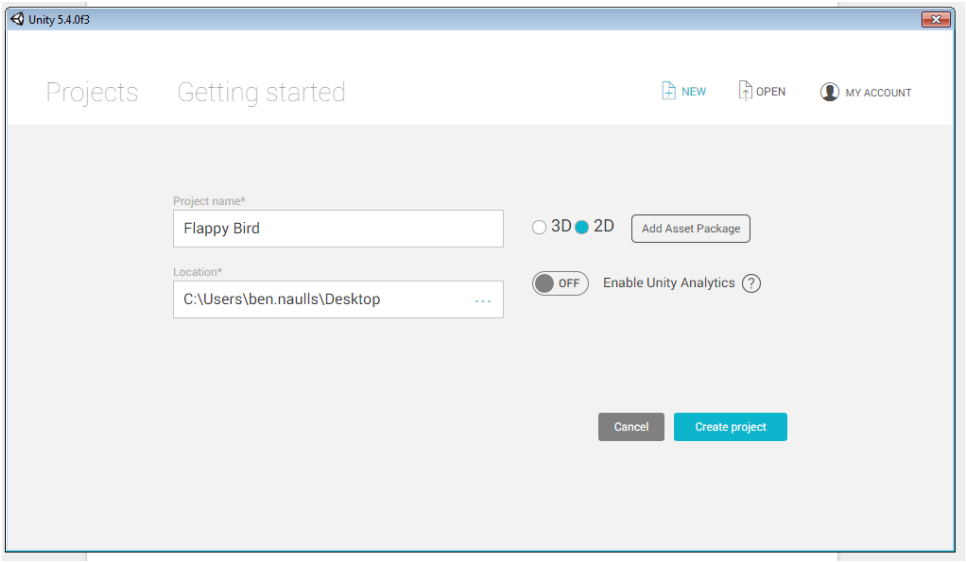
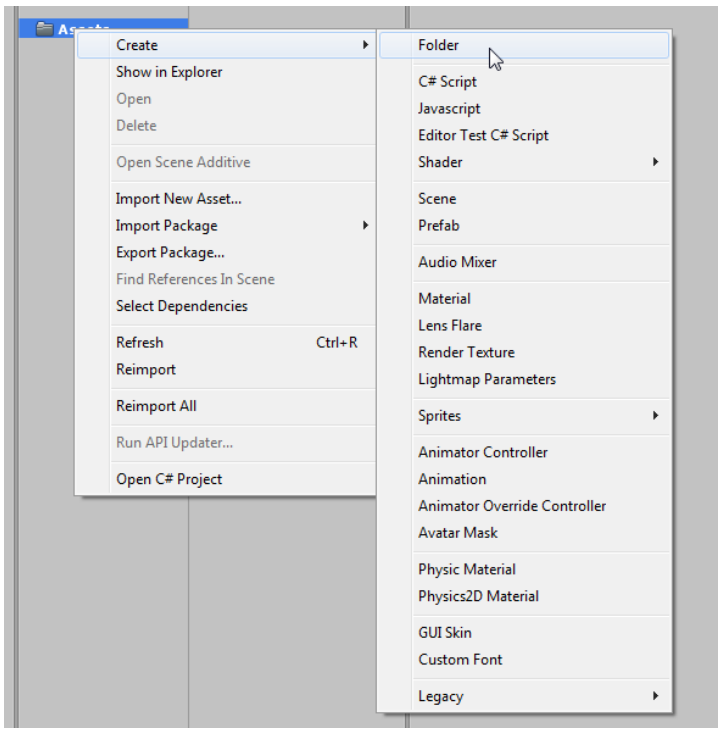
Unity: Making “Flappy Bird”

Starting the Project

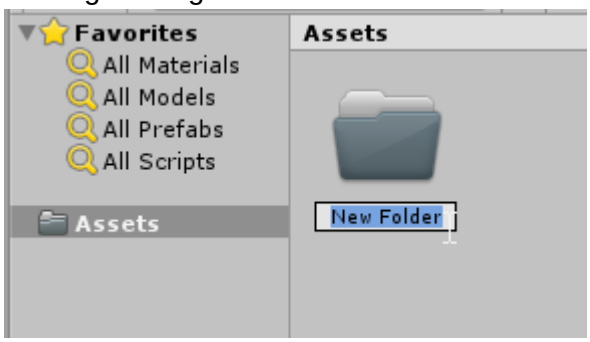
Create a new project and ensure the project is set up as 2D (these instructions won't work if you skip this step)

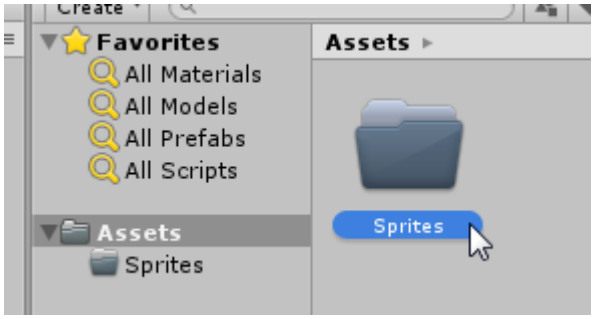


Right Click on the **Assets** folder in the Project view and select **Create > Folder**



Rename the folder to **Sprites** by clicking on the name of the folder waiting 2 seconds and clicking on it again.

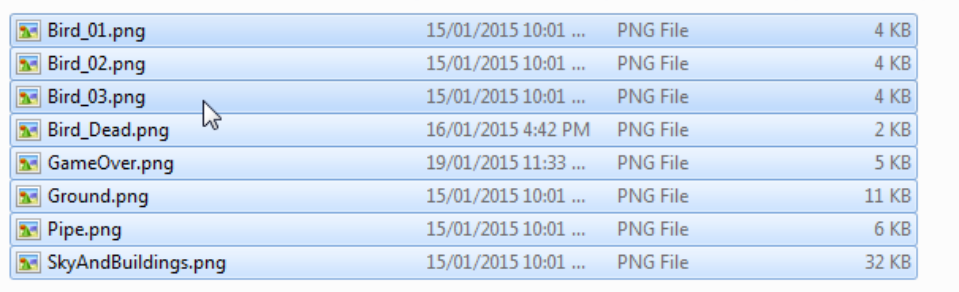




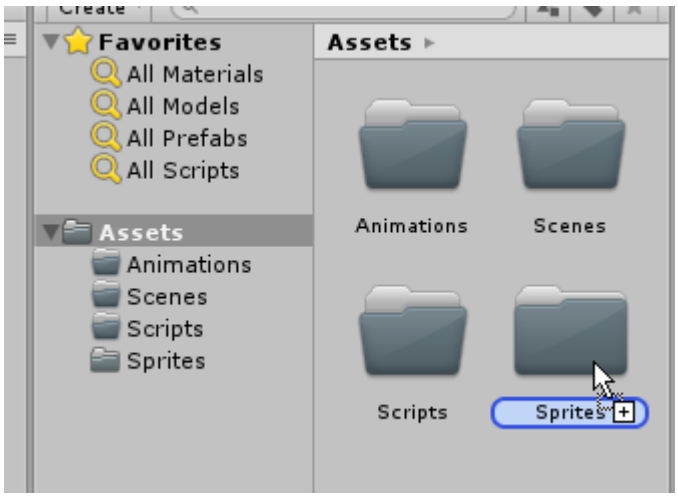
Repeat this step and create a folder for:

- **Scripts**
- **Animations**
- **Scenes**
- **Fonts**

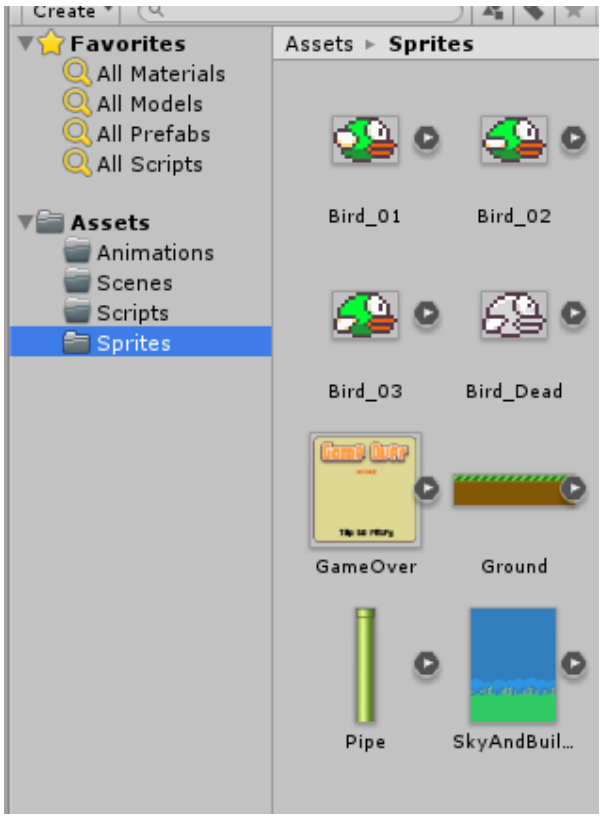
You have been provided with all the sprites for the game, select them all

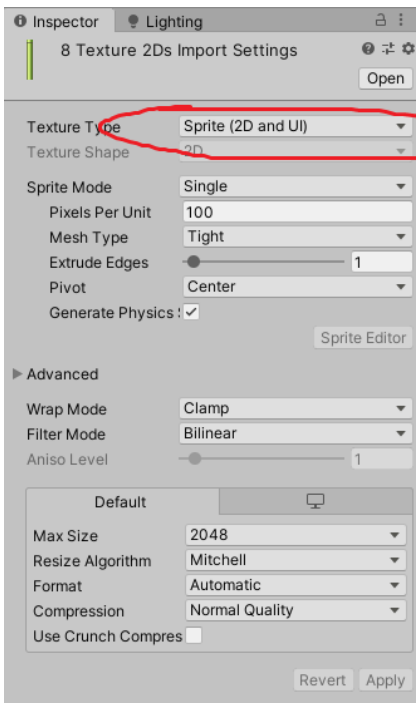


And drag them into the new **Sprites** folder



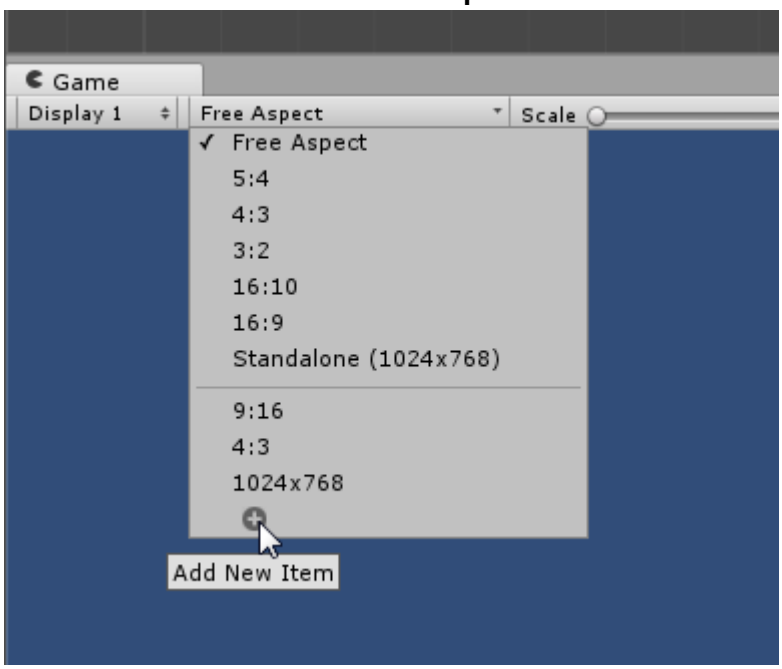
Make sure you set the Texture Type to Sprite in each images inspector options.



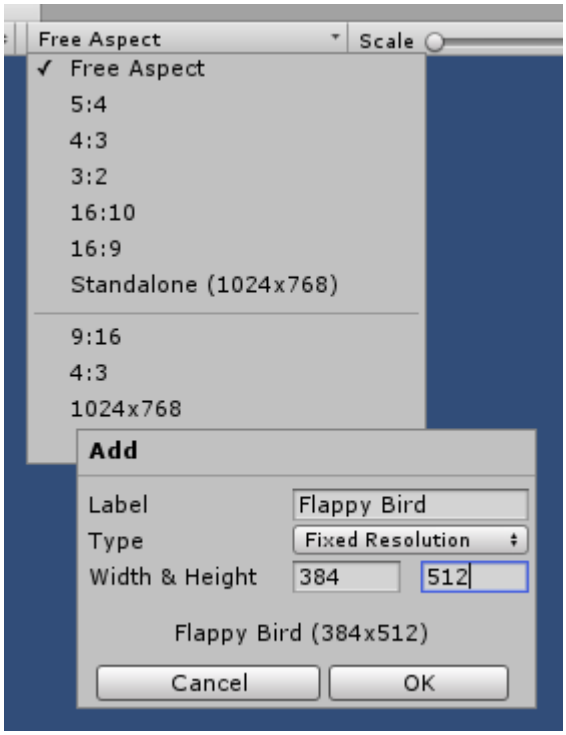


Setting up the scene

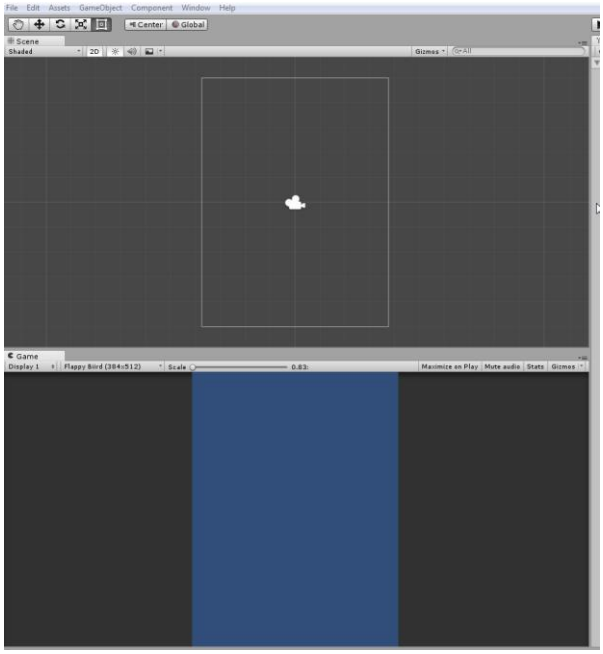
In the **Game** view click the **Free Aspect** button and select the **+** from the dropdown view.



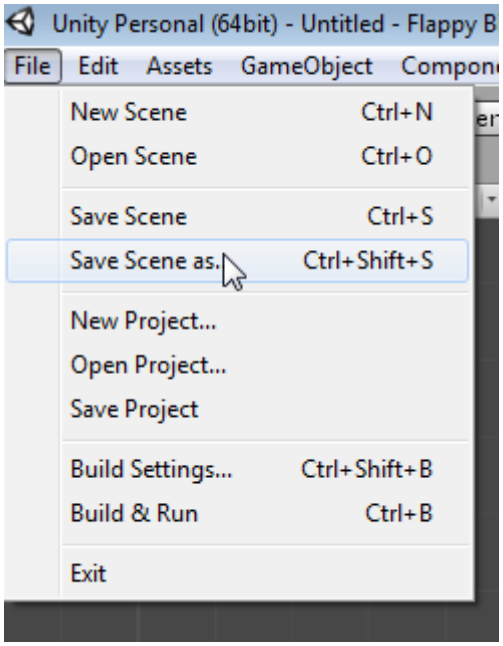
In the window that pops up change the label to **Flappy Bird** and the Width and Height to **384** and **512**



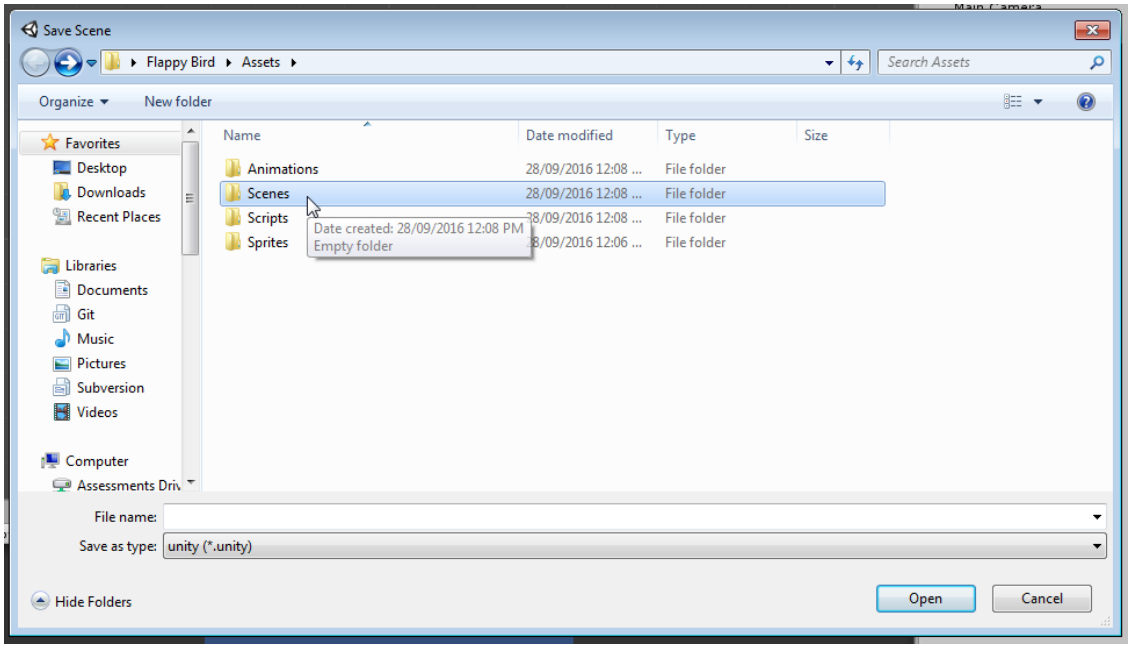
Once you click OK you will see the window change to a tall view.



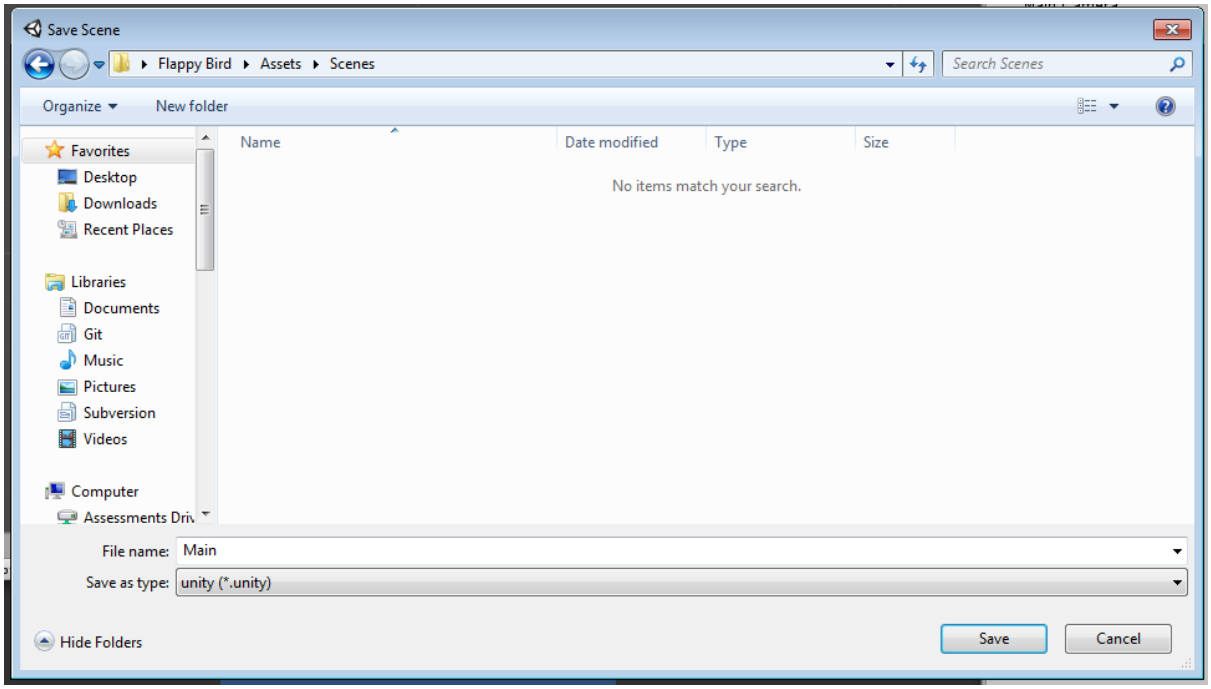
Save the scene by going to **File > Save Scene as...**



In the window that pops up pick the scenes folder

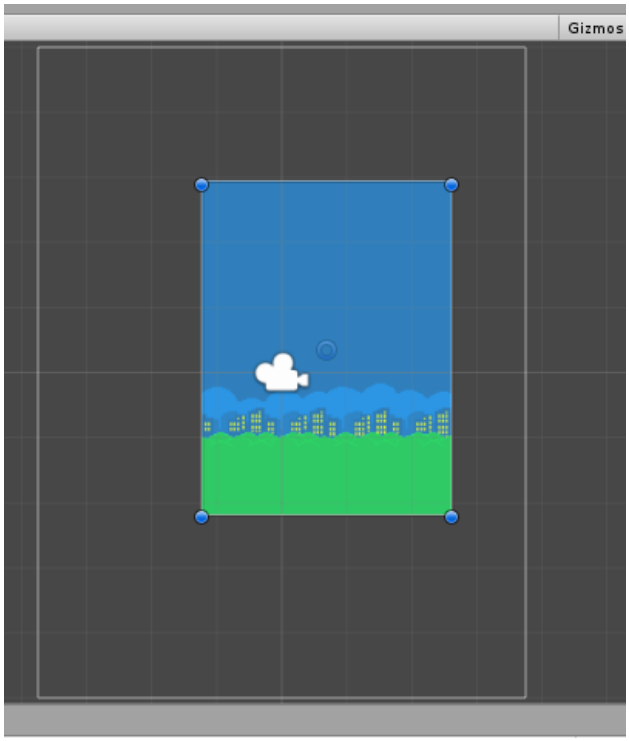


Then name the file **Main** and click the **Save** button

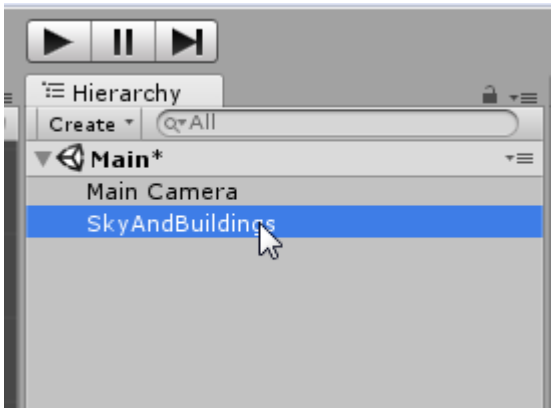


Putting the background in

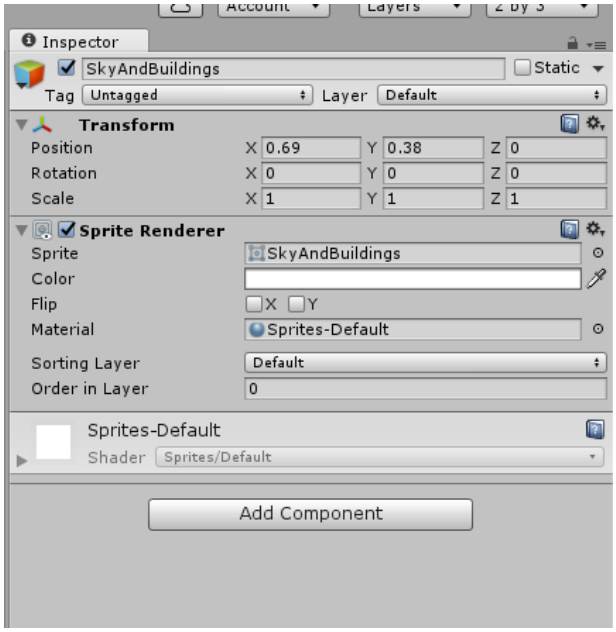
Drag the **SkyAndBuildings** sprite from the **Project** view to the **Scene** view



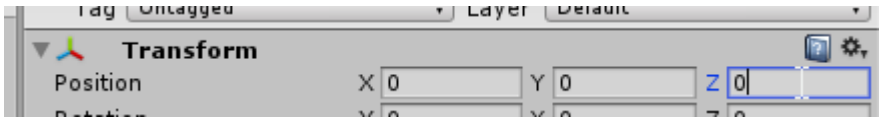
We need to center this and match the camera its size.
First with the **SkyAndBuildings** selected in the **Hierarchy** view



You should see something like this in the **Inspector**

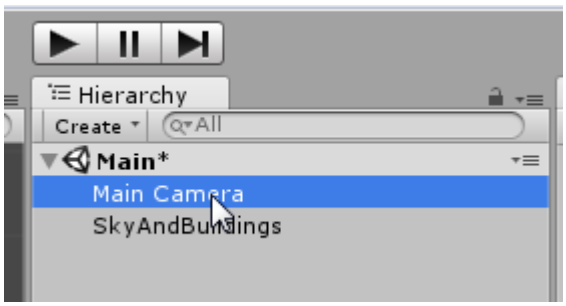


Under Transform, **Position** tells us where on the screen this sprite will sit. Change the X, Y and Z to all read **0**

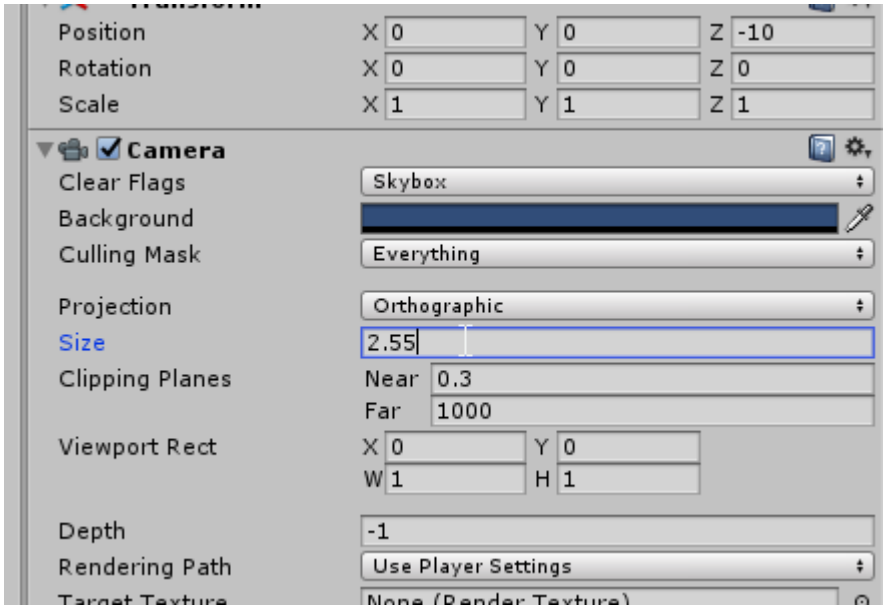


You should see the background image center in the **Scene** view

Click on the **Main Camera** in the Hierarchy view



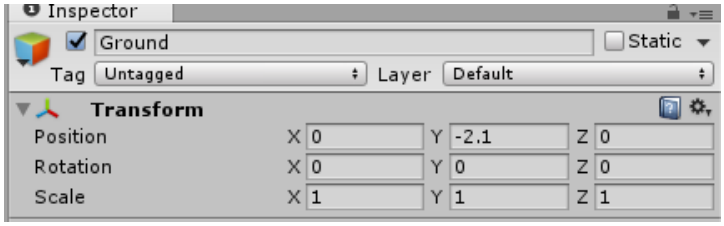
And in the **Inspector** view change the **Size** value to read **2.55**

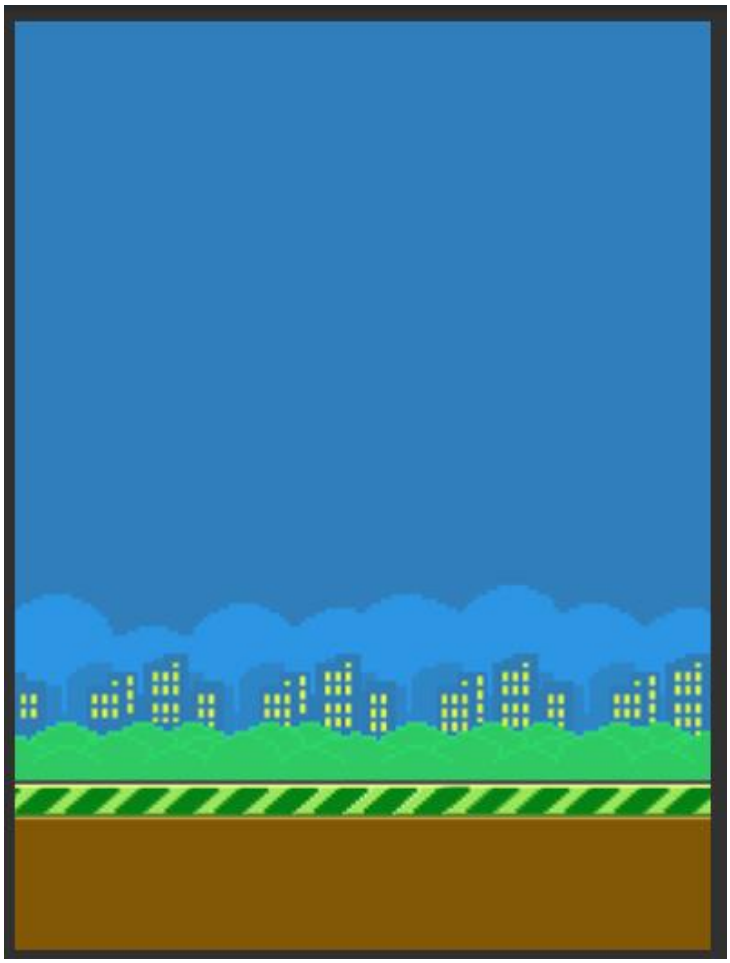

Position
X 0
Y 0
Z -10
Rotation
X 0
Y 0
Z 0
Scale
X 1
Y 1
Z 1
Camera
Clear Flags
Skybox
Background
Culling Mask
Everything
Projection
Orthographic
Size
2.55
Clipping Planes
Near 0.3
Far 1000
Viewport Rect
X 0
Y 0
W 1
H 1
Depth
-1
Rendering Path
Use Player Settings

Adding Ground

Drag the **Ground** sprite to the **Scene** view

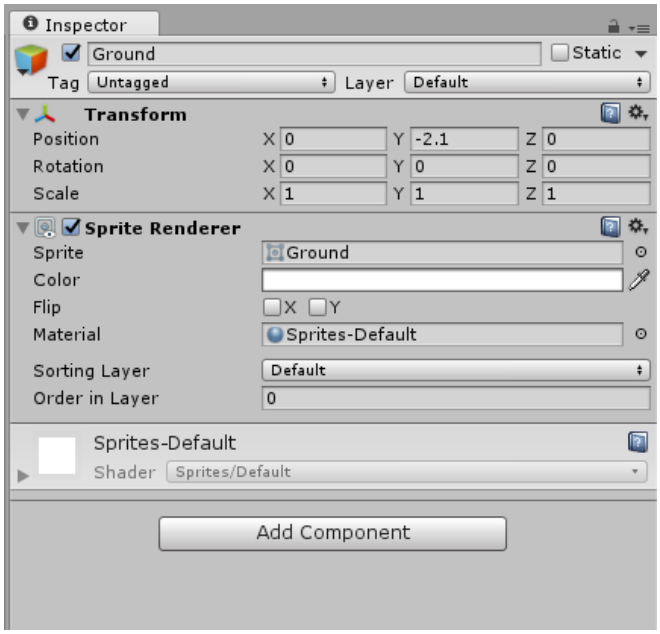
Then change its position to
X: 0
Y: -2.1
Z: -1





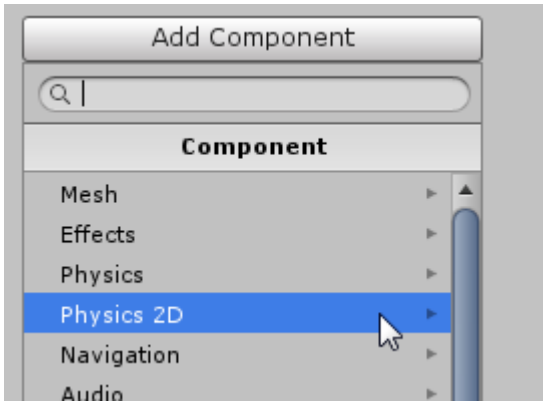
We now need to make a collider on the ground object so that Unity knows when something collides with it.

Select the **Ground** object in the **Hierarchy** view and in the **Inspector** view you should see this

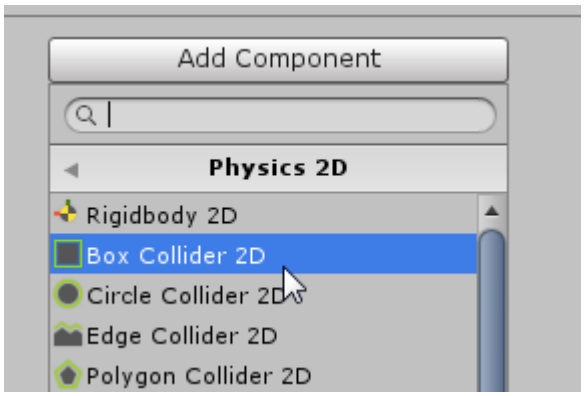


Click on the **Add Component** button to add things to this object. What we need to add is under

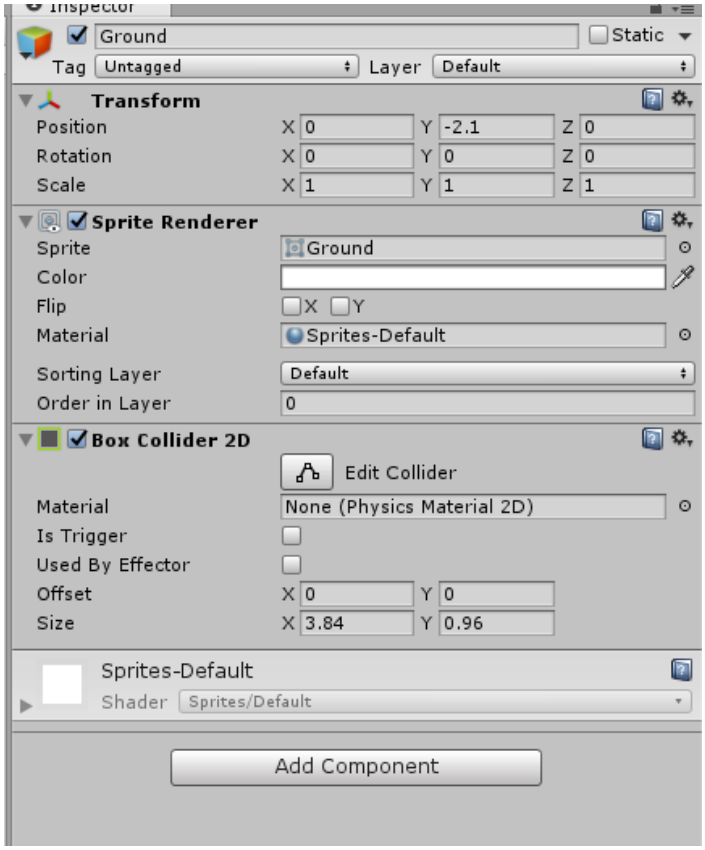
Physics 2D



Box Collider 2D



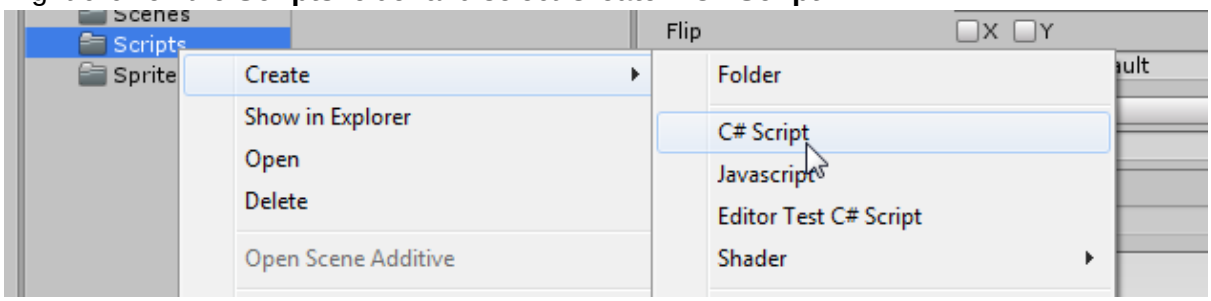
You should now see the box collider attached to the ground object



Now we need a script to make the ground move

In the **Project** view

Right click on the **Scripts** folder and select **Create > C# Script**



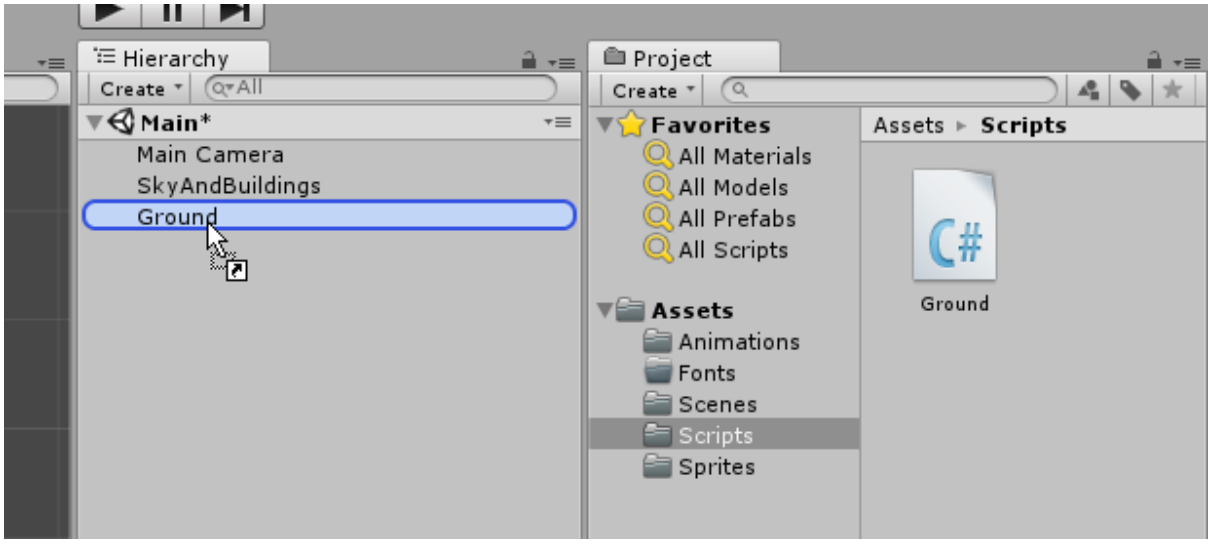
Name this script **Ground**

Double click on this file and **Visual studio** will load. This will take a while the first time you do this.

Type this code into the editor and save it with **Ctrl + S**

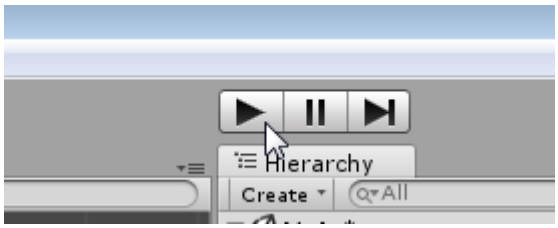
```
using UnityEngine;
using System.Collections;

public class Ground : MonoBehaviour
{

    public float speed = 1.5f;

    // Use this for initialization
    void Start()
    {

    }

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

        // Every frame we look at the position of the ground and move it left ever so slightly
        transform.position = transform.position - (Vector3.right * speed * Time.deltaTime);

        // If the position of the ground is off the left of the screen....
        if (transform.position.x <= -3.84f)
        {

            // Move it to the far right of the screen
            transform.position = transform.position + (Vector3.right * 3.84f * 2);
        }
    }
}
```

Back in Unity drag the **Ground** script you just created from the **Project view** to the **Ground** object in the **Hierarchy** view. This will tell unity to make the ground object use the ground script.



Once this is done we are ready to play

Press the **Play** button at the top of the window

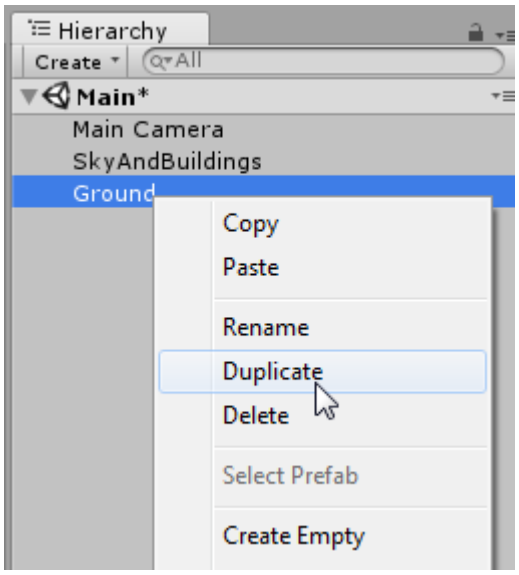


If done correctly the ground object will scroll to the left until it is off the side of the screen and then move to the far right of the screen and continue.

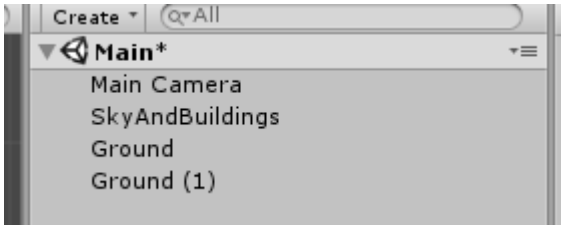
But you will notice that it leaves a large hole. Let's fix this by making a second ground piece.

Stop the game by clicking the **Play Buttons** again

In the **Hierarchy** view right click on the **Ground** object and select **Duplicate**



You should now see a second ground object in the **Hierarchy** view



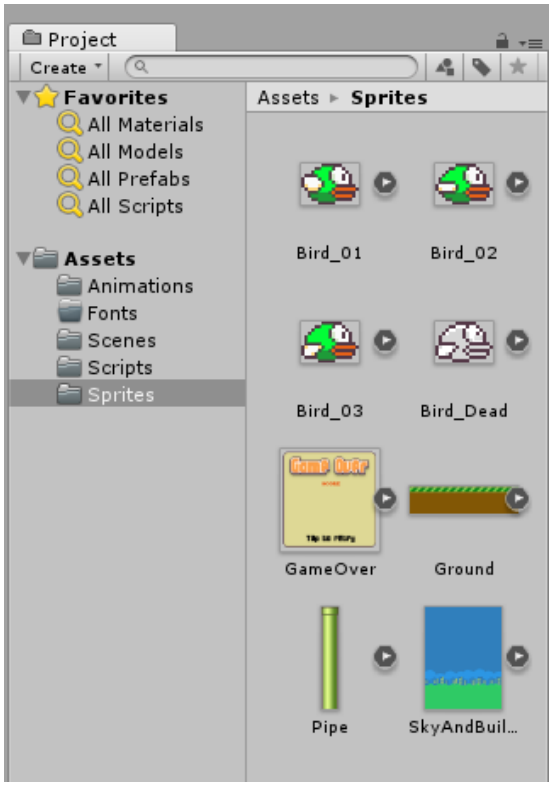
Select the new ground object and in the **Inspector** change it's position to

X: 3.84
Y: -2.1
Z: -1

Press **Play** again and see what happens

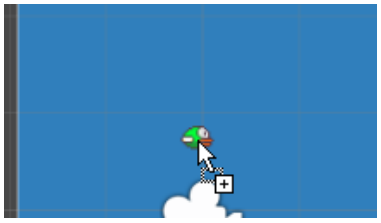
Adding the bird

Open up your **Sprites** folder

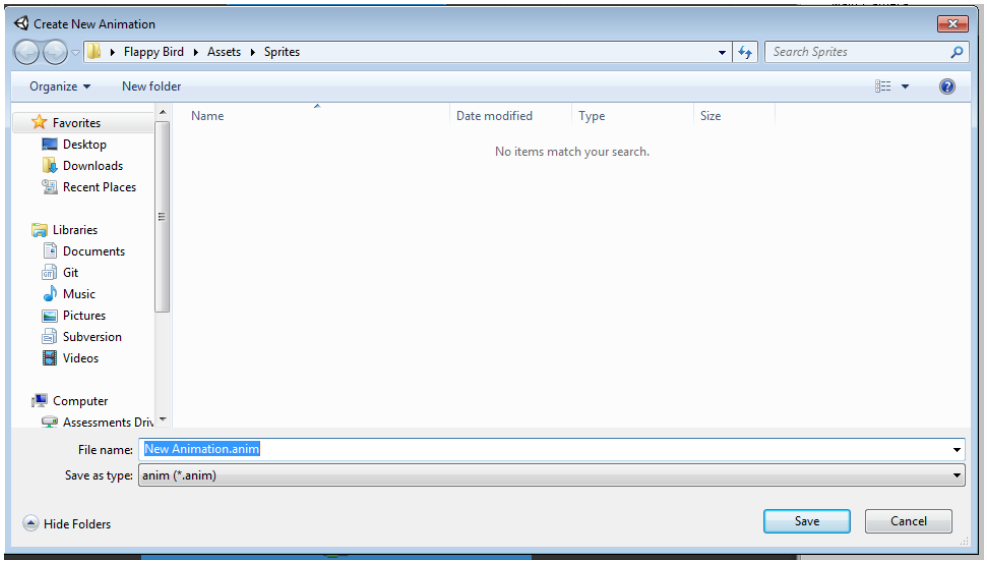


While holding down **Shift** select **Bird_01, Bird_02, and Bird_03**

Then drag all three into the **Scene** view

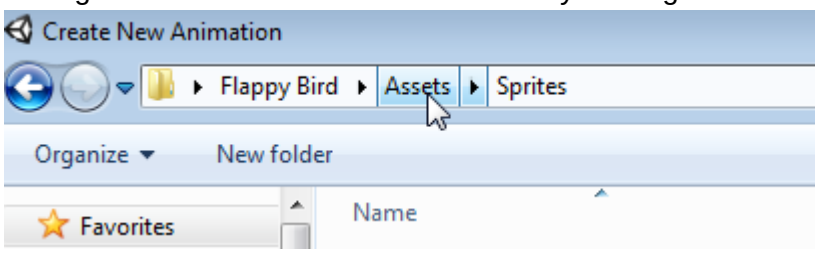


When you do this you should see this window

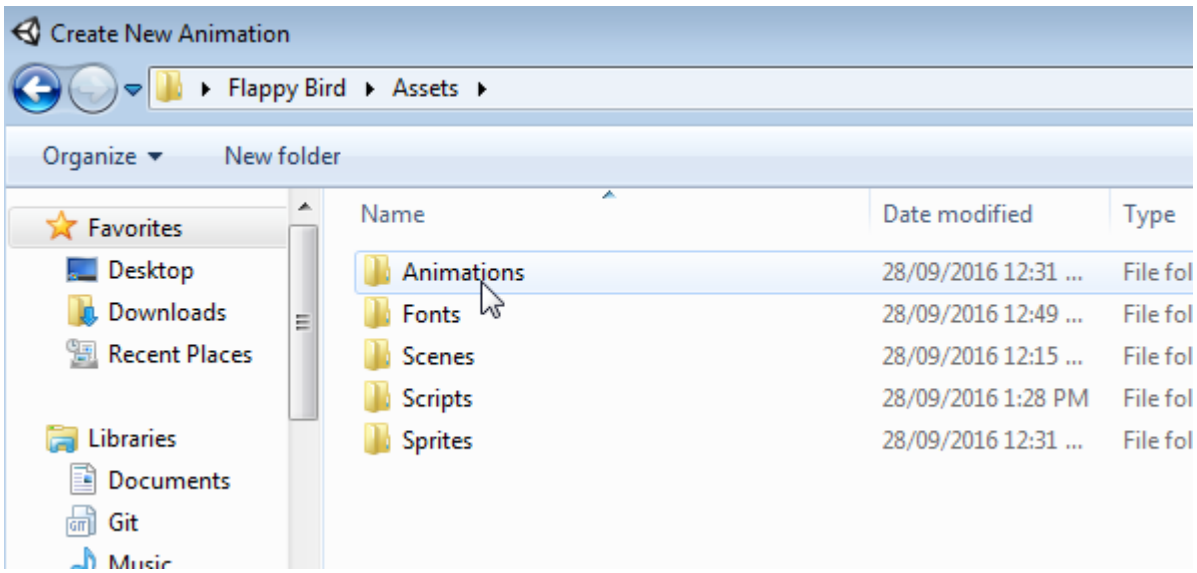


Unity detects that you are dragging in multiple sprites and is trying to make an animation with them

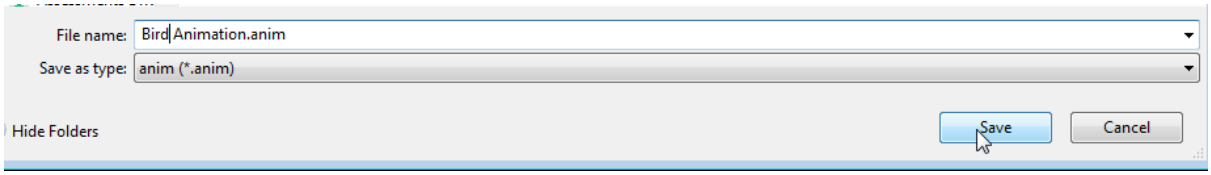
Change the folder to the animation folder by clicking on **Assets** on the file path



Then clicking on the **Animations** folder



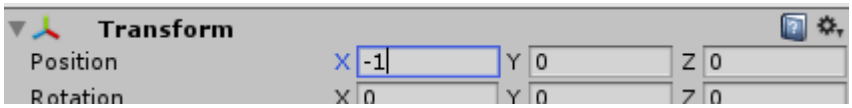
Then call the file name **Bird Animation.anim**



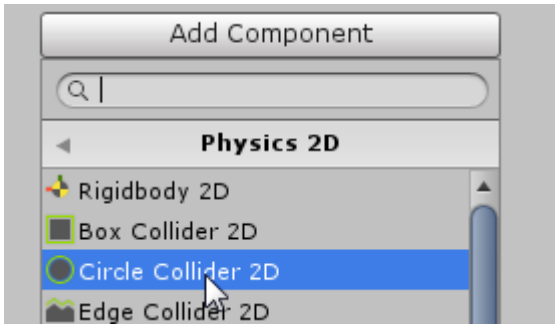
And click the **Save** button.

Select **Bird_02** from the **Hierarchy** and in the **Inspector** change its position to

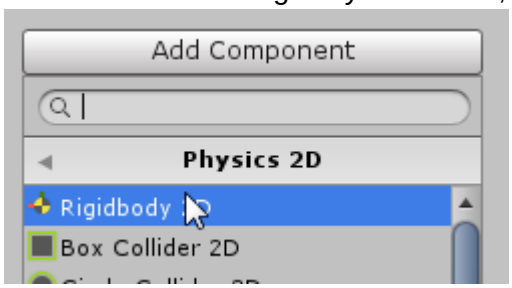
X: -1
Y: 0
Z: -1



Then add a **Circle Collider** Component like we did with the **Ground** object



And we need to add gravity to the bird, you do this by adding a **RigidBody 2D** Component



Press play and see what happens

Your flapping bird should fall and hit the ground
Let's Make a new script in the scripts directory called **Bird**

Copy the following code into the script and save with **Ctrl+S**

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

public class Bird : MonoBehaviour
{

    public float flapHeight = 5;

    // Use this for initialization
    void Start()
    {

    }

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

        if (Input.GetKeyDown(KeyCode.Space))
        {

            //Stop the bird moving downwards
            GetComponent<Rigidbody2D>().velocity = Vector3.zero;

            //Push the bird up with a force that equils the flapHeight
            GetComponent<Rigidbody2D>().AddForce(Vector2.up * flapHeight, ForceMode2D.Impulse);
        }
    }

    void OnCollisionEnter2D()
    {
        SceneManager.LoadScene(0);
    }
}
```

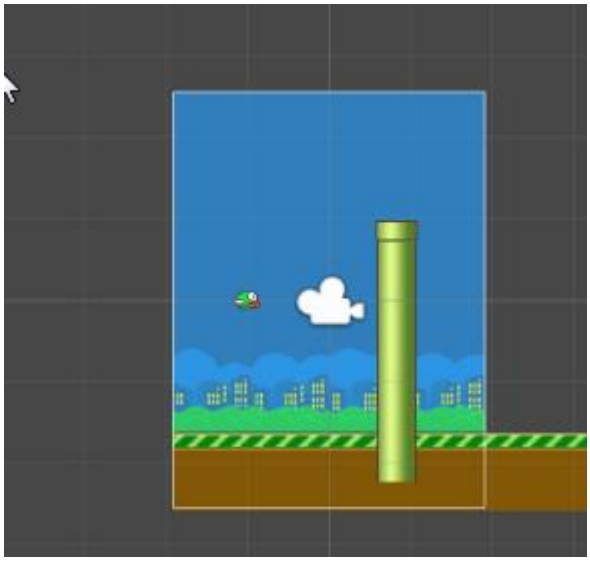
Drag this script from the Project View onto the **Bird** object in the **Hierarchy** view.

Now press play and you should be able to control the bird with the space bar.

Add Pipes

Now we are going to add the pipes, We add our first pipe the same way that we added the ground.

Drag the **Pipe** sprite into the **Scene** view



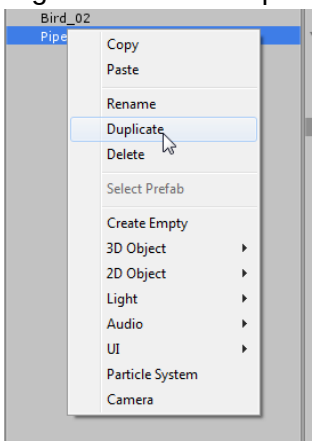
Change the position of the pipe to:

X: 1
Y: -1
Z: -1

And add a **Box Collider2D** to it like we did with the ground

Now since we need two pipes (one of top, one on the bottom) we need to duplicate the pipe we just created.

Right click on the Pipe in the **Hierarchy** view and click on **Duplicate**



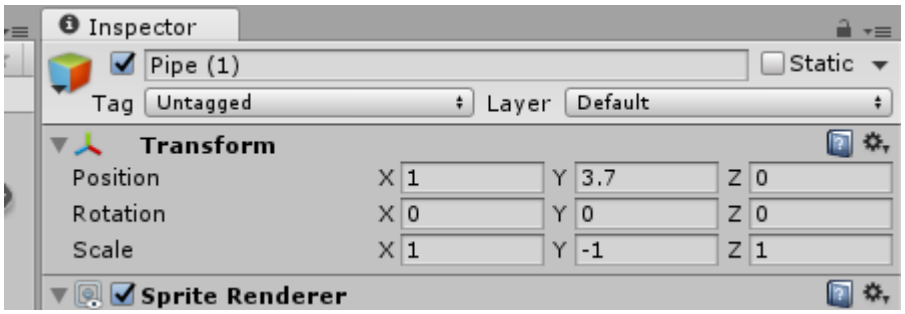
You should now have two Pipes in the **Hierarchy** view

Select the second Pipe and in the **Inspector** view change the **Position** and the **Scale** to:

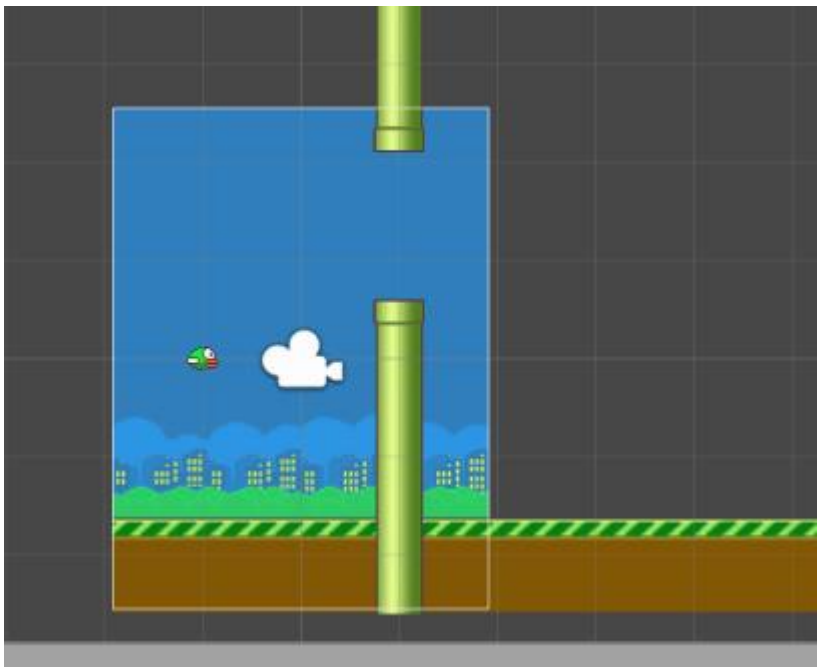
Position:

X: 1
Y: 3.7
Z: -1

Scale
X: 1
Y: -1
Z: 1



Your scene should now look like this



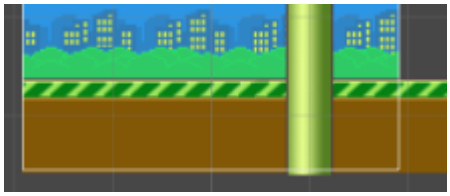
To get the Pipes to move together we are going to Parent one to the other.

Currently your Pipes are separate objects

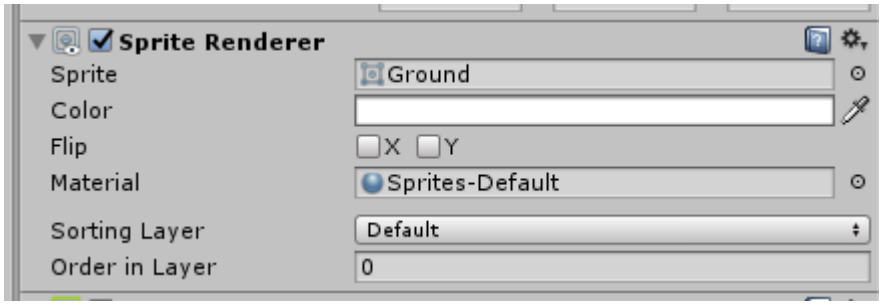
In the **Hierarchy** view drag **Pipe (1)** into **Pipe**

This will make it so when **Pipe** moves **Pipe (1)** will move with it

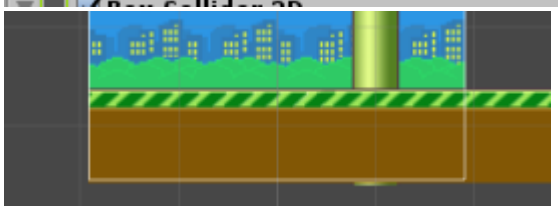
You might have noticed that the pipe is drawn over the ground, we can fix this by changing the layer order of Items.



Select one of the ground object look at the **Inspector** view



The **Order in Layer** tells Unity which object to draw onto of others.
Change the number from **0** to **1**



Do this with the other ground object

New we need to make a script to control the pipes

Create a new script called **Pipe** in the **Scripts** folder and copy in the following code:

```
using UnityEngine;
using UnityEngine.UI;
using System.Collections;

public class Pipe : MonoBehaviour {

    public float speed = 1.5f;
    private int score = 0;

    // Use this for initialization
    void Start () {
```

```
        }

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

                // Every frame we look at the position of the ground and move it left ever so
slightly
                transform.position = transform.position - (Vector3.right * speed *
Time.deltaTime);

                // If the position of the ground is off the left of the screen....
                if (transform.position.x <= -2.3f) {

                        // Move it to the far right of the screen
                        transform.position = transform.position + (new Vector3(2.3f*2,0,0));
            transform.position = new Vector3(transform.position.x, Random.Range(-2.5f, -1),
0);
            if (GameObject.Find("Text")){
                GameObject.Find("Text").GetComponent<Text>().text = (++score).ToString();
            }

        }
        }
}
```

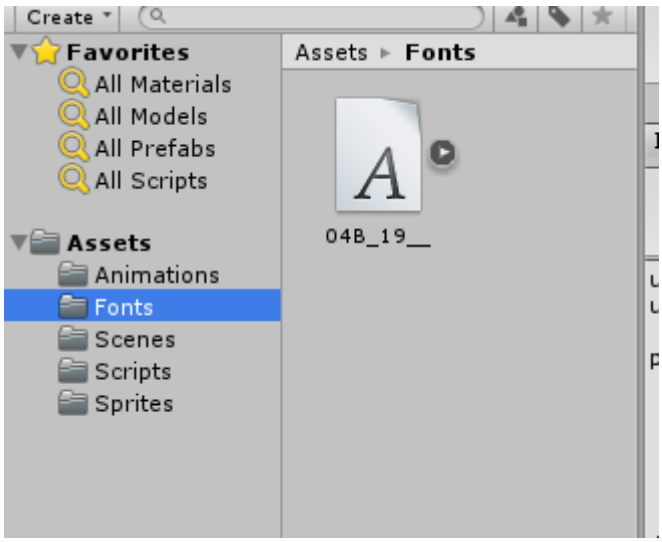
Save the scripts with **Ctrl + S**

Then drag the script from the **Project** view to **Pipe** in the **Hierarchy** view (not **Pipe (1)**)

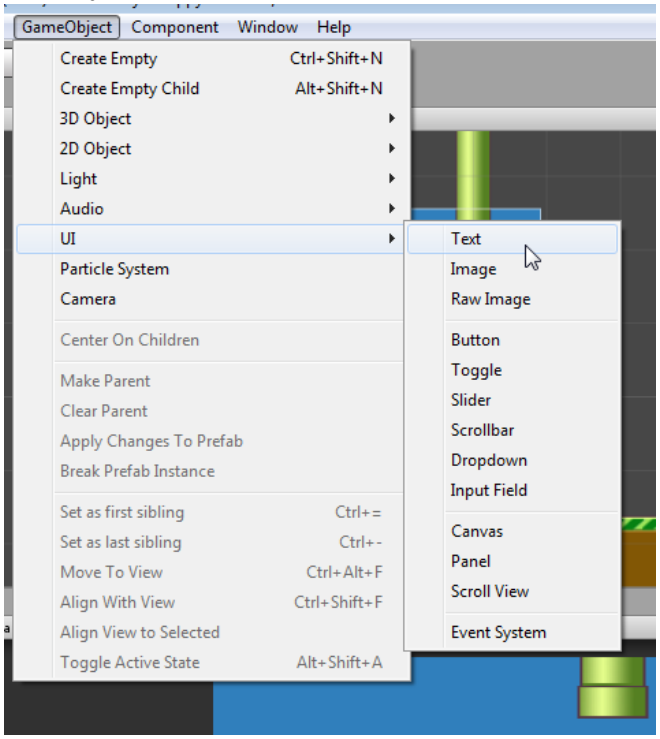
Press Play as see what happens

Adding the UI

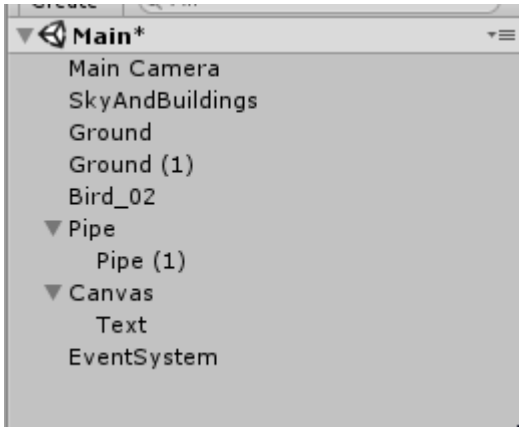
You have been given a font file called **04B_19__.TIF**, Drag this into your **Fonts** folder



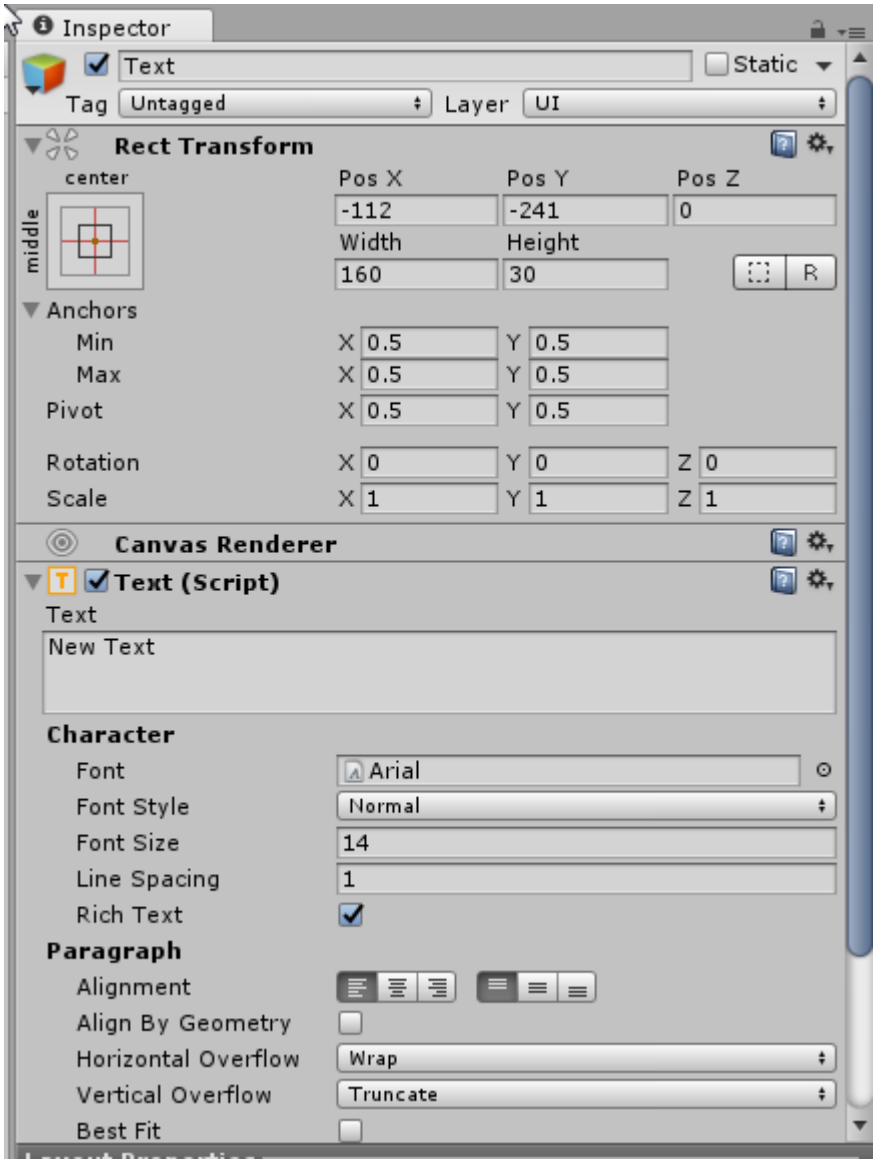
In Unity's menu select **GameObject > UI > Text**



This should create several new items in the **Hierarchy** view



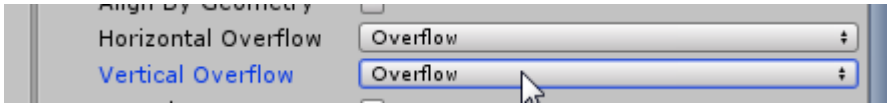
Select the **Text** object and look at the inspector



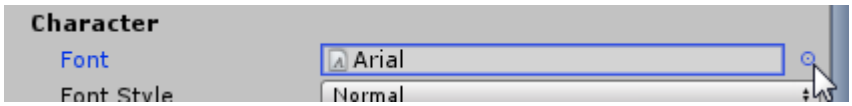
Here we have a lot of options, Let's change the following:

Change the colour of the font to **White**

Change both **Horozontal** and **Vertical Overflow's** to **Overflow**



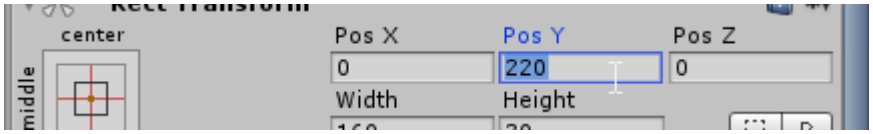
Lets select our custom font, Click on the little circle next to **Font**



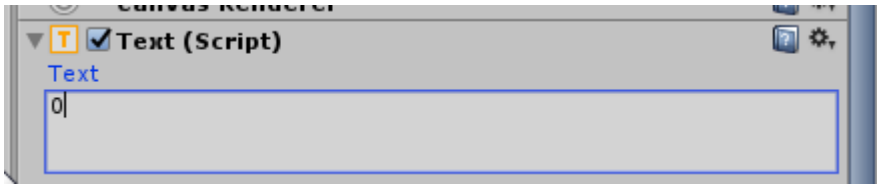
Font Size to 50

Alignment to Center

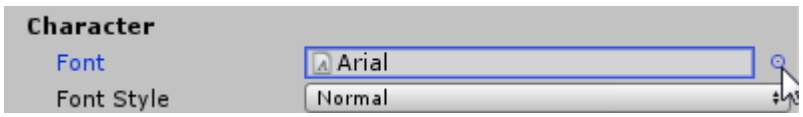
Rect Transform position to
Pos X: 0
Pos Y: 220
Pos Z: 0



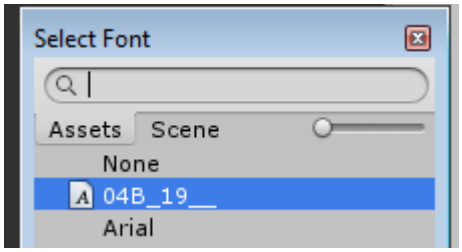
Text to "0"



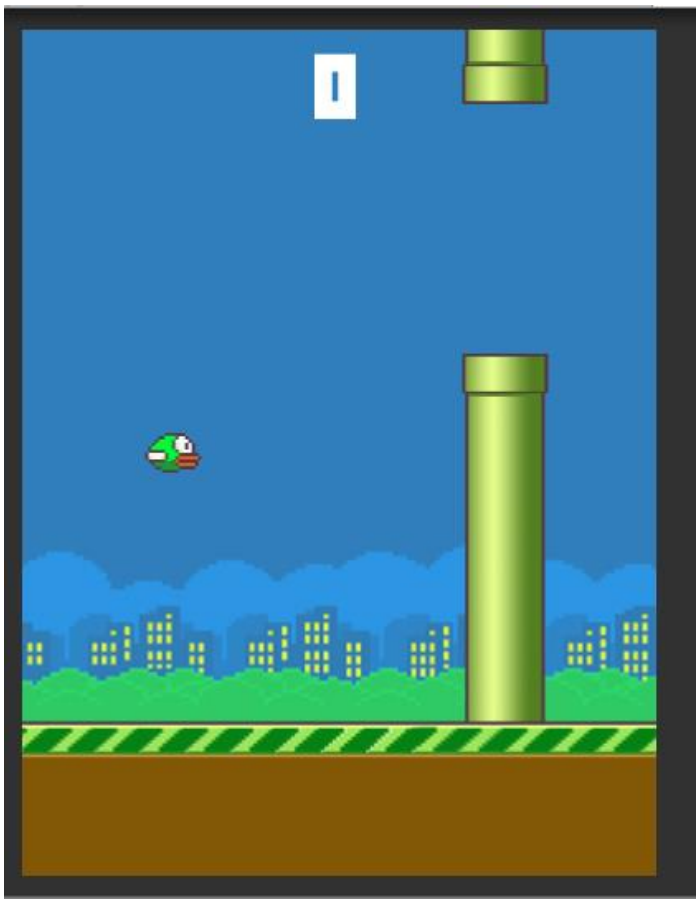
Click on the little circle next to the **Font**



In the window that pops up select our custom font **04B_19__**



Your game should now look like this:



Press play and see what happens.