

UIT2627 - INTRODUCTION TO AR/VR/MR/XR

AIM:

The primary aim of this project is to develop a replica of the classic "Flappy Bird" game using the Unity game engine. This project aims to familiarize participants with essential Unity components, physics, and scripting concepts.

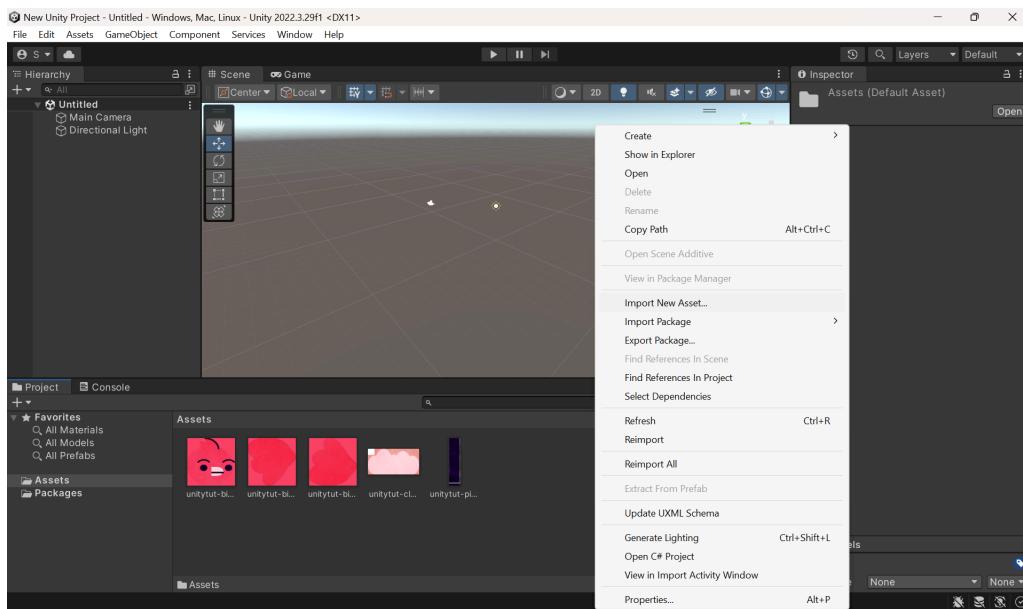
OBJECTIVES:

1. **Understanding the Unity Interface:** Gain proficiency in navigating and utilizing various Unity panels such as Project, Hierarchy, Inspector, and Scene.
2. **Implementing Game Physics:** Apply physics to the bird object to simulate realistic gravity and motion.
3. **Creating Spawning Objects:** Develop dynamic spawning of pipes that move continuously across the screen, providing obstacles for the player.
4. **Scripting Game Logic:** Write custom scripts to control the bird's movement and the spawning and destruction of pipe objects.
5. **UI Development:** Implement a user interface to display the game score and other relevant information.
6. **Colliders and Interaction:** Use colliders to detect interactions between game objects and trigger game events.

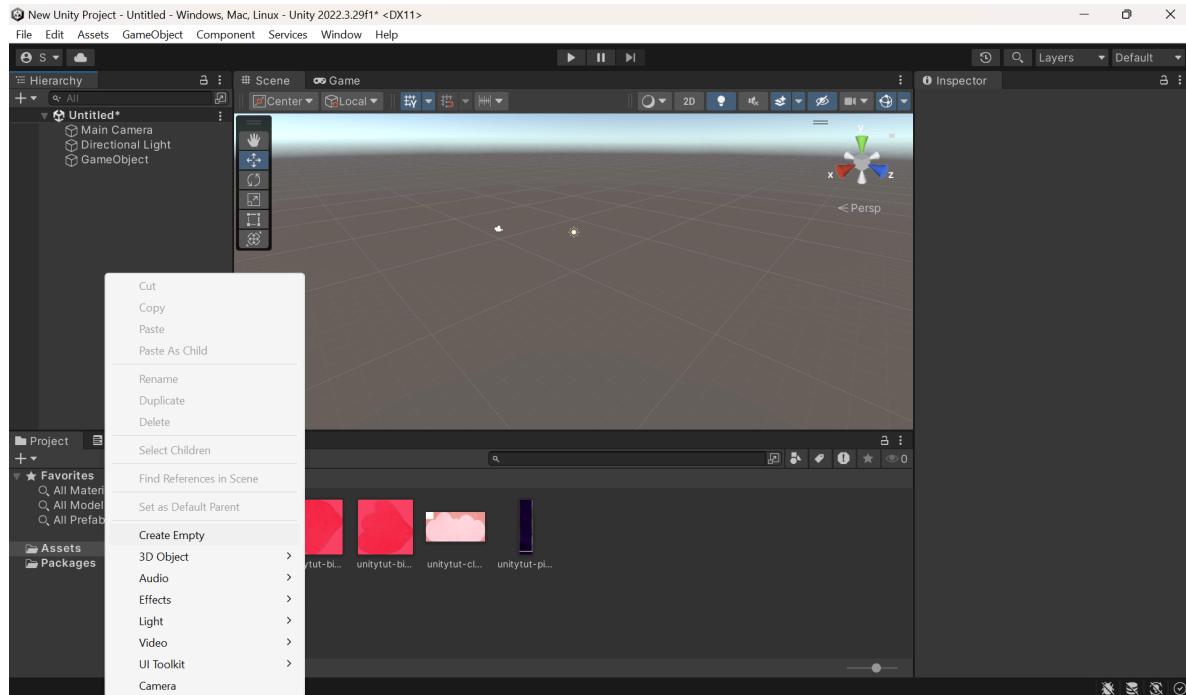
STEP BY STEP PROCEDURE:

Step 1: Getting Familiar with Unity Interface

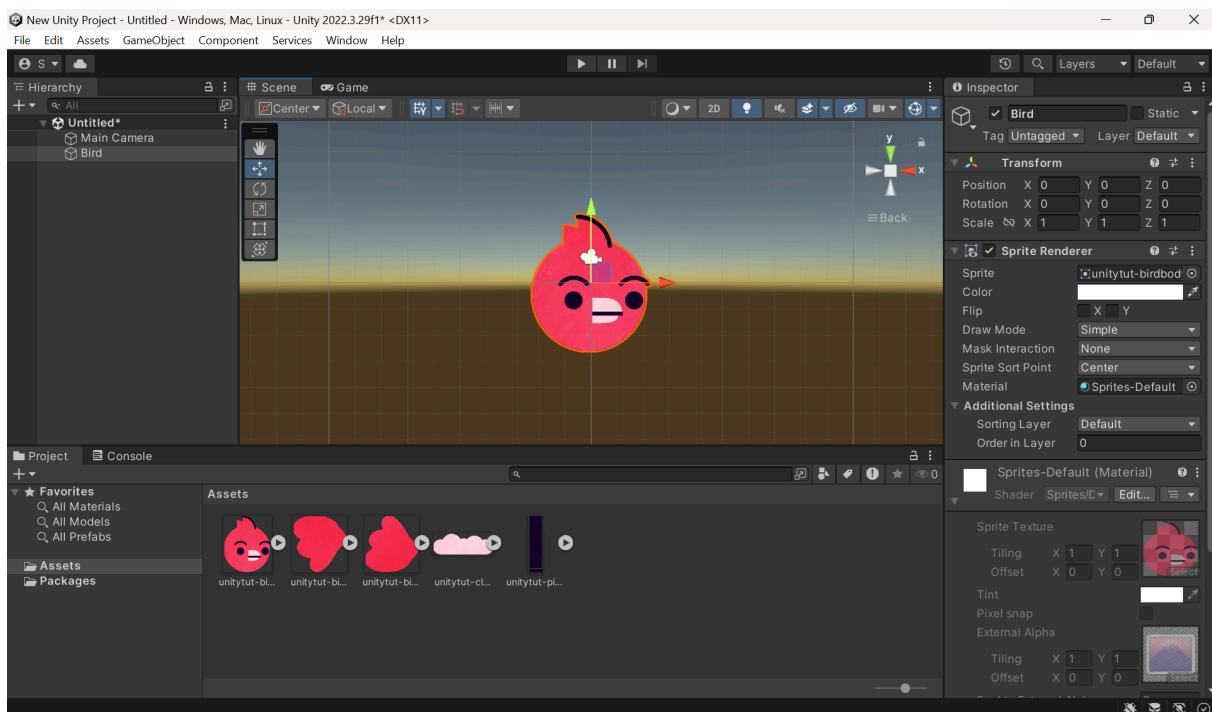
1. The first panel is named Project. Download the assets, by clicking on Asset => Import New Asset => Select the required assets and click on Import



2. The second panel is called hierarchy, right click and select Create Empty => A GameObject is created, it is an invisible container which has position in space, a rotation and scale. We can also fill the container with components to add extra features - Bird, Pipe, UI, Camera.



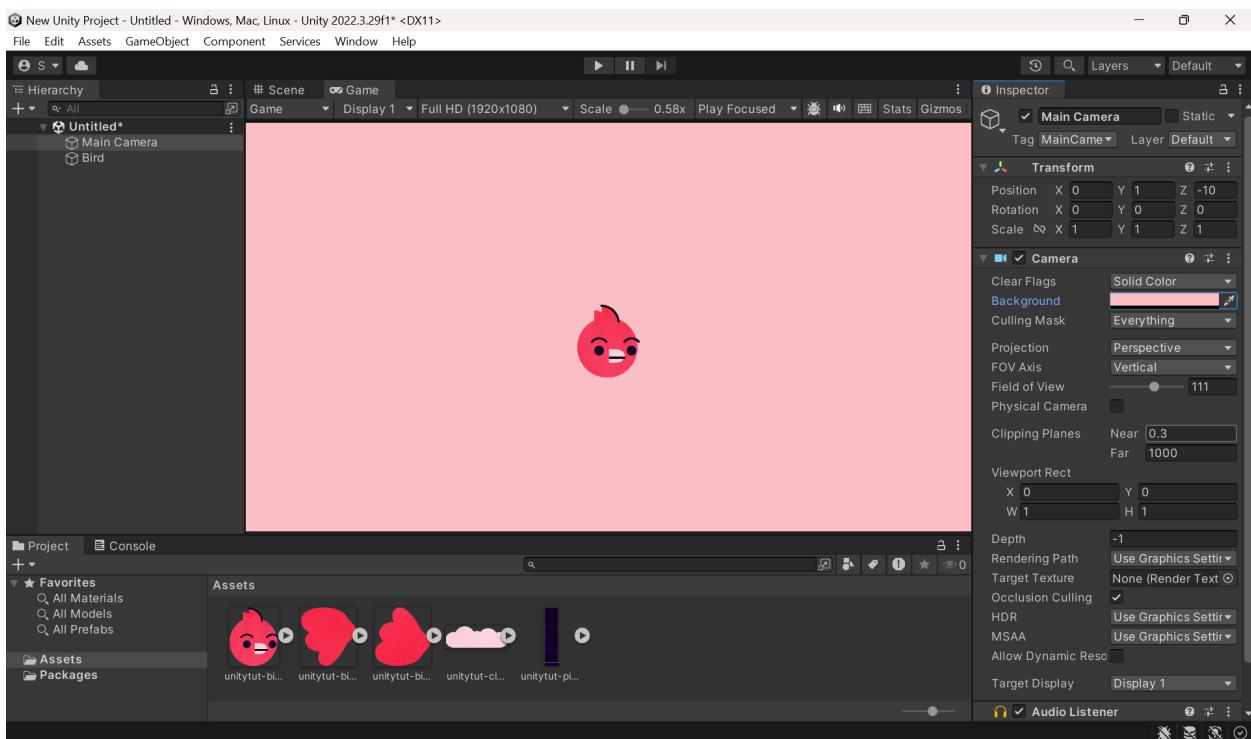
3. The third panel is called Inspector, where we can manipulate the GameObjects. Now we select the game object and rename it as Bird, click on Add Component => Rendering => Sprite Renderer => Drag and drop the bird asset into the Sprite Renderer



4. The fourth panel is named Scene. This is the game view.

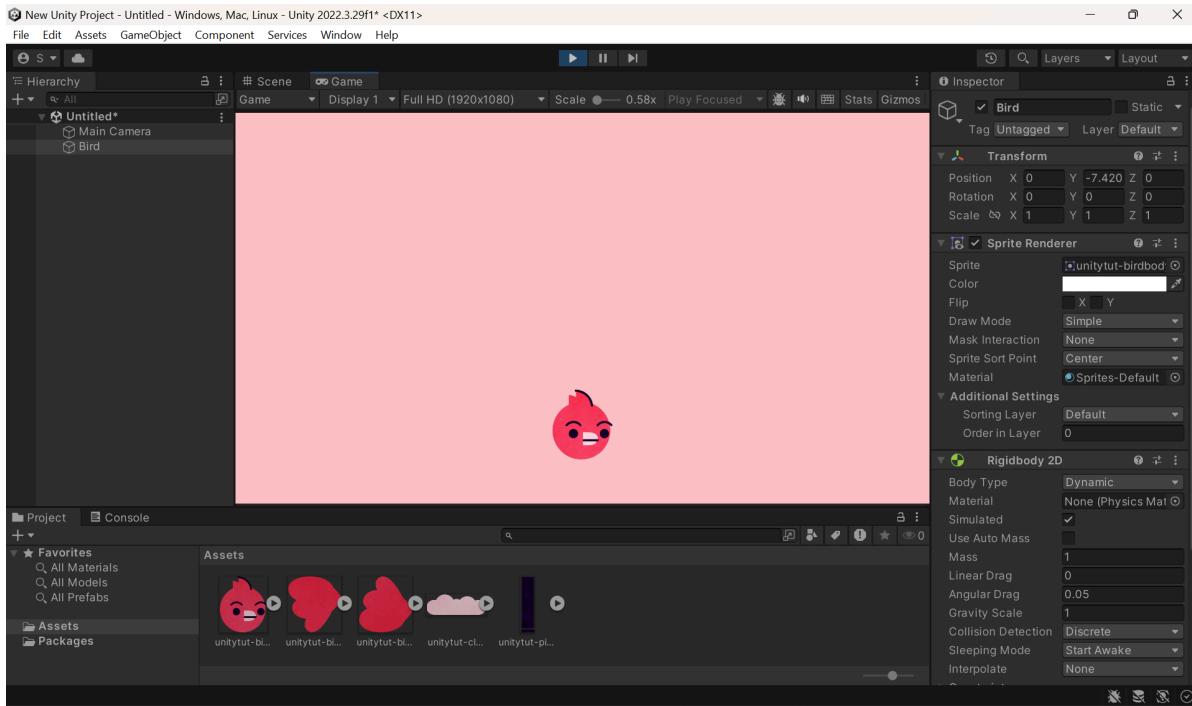


5. Now, I click on the Main Camera Object and change the background color and increase the field of view.

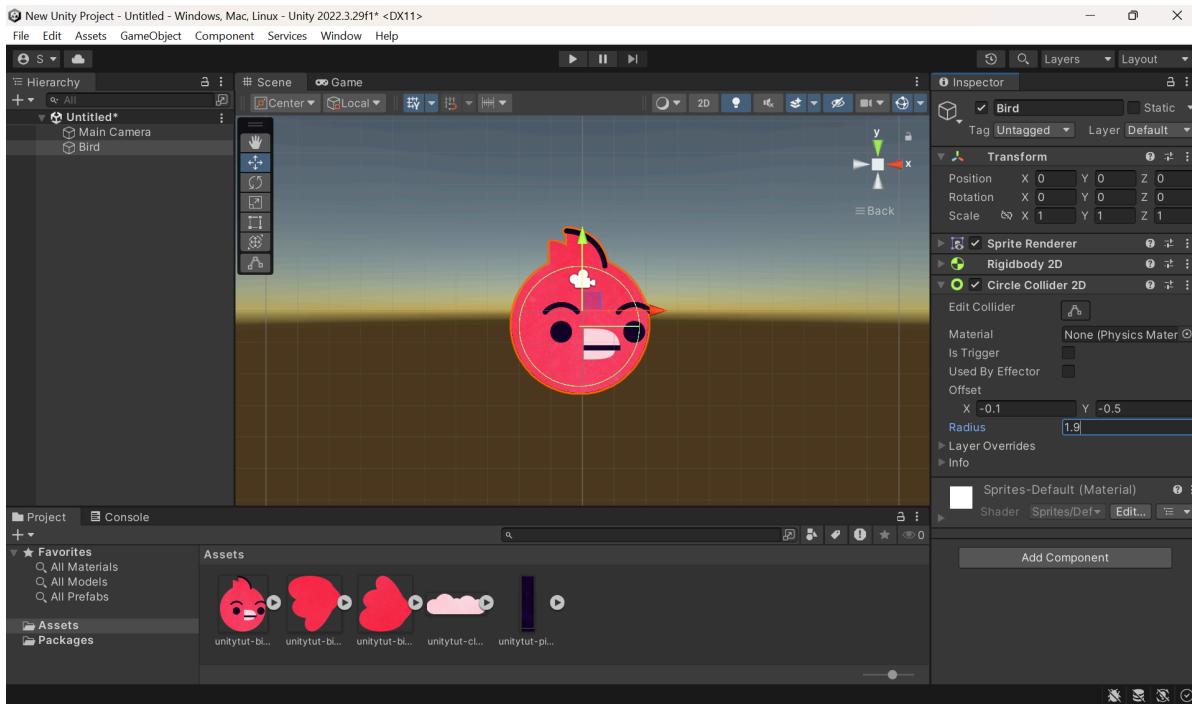


Step 2: Physics and Programming to Move the Flappy Bird when we press space bar

1. Adding another component to the bird - a Rigidbody 2D - This turns the bird into a physics object with gravity. So when we hit play the bird drops and falls off the screen.
Bird => Add Component => Physics 2D => Rigidbody2D => Hit play => bird drops down



2. Now to make the bird interact with other objects, we add a collider, Add Component => Search Collider => Circle Collider 2D => I change the offset and radius to fit the collider inside the bird.



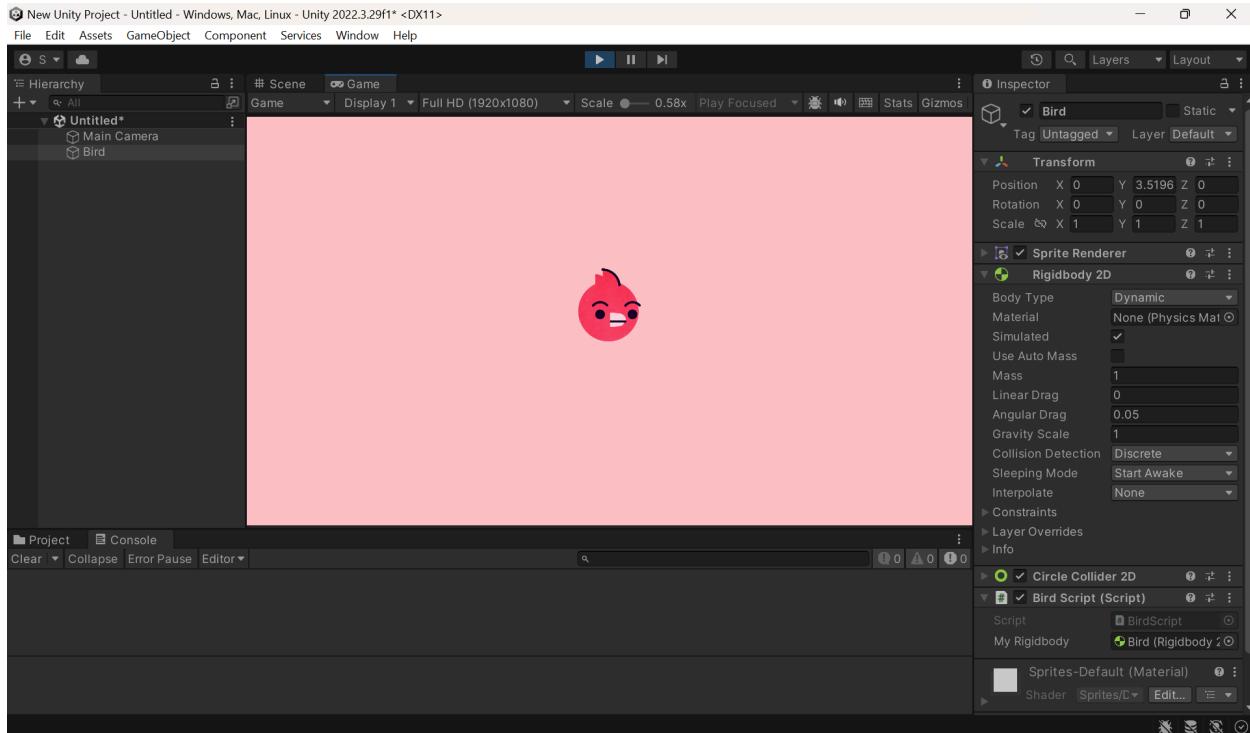
3. Now we create our own custom script. Add Component => New Script => Name it as Bird Script => Select My Rigidbody as Bird and enter the following code.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class BirdScript : MonoBehaviour
{
    public Rigidbody2D myRigidbody;
    void Start()
    {

    }

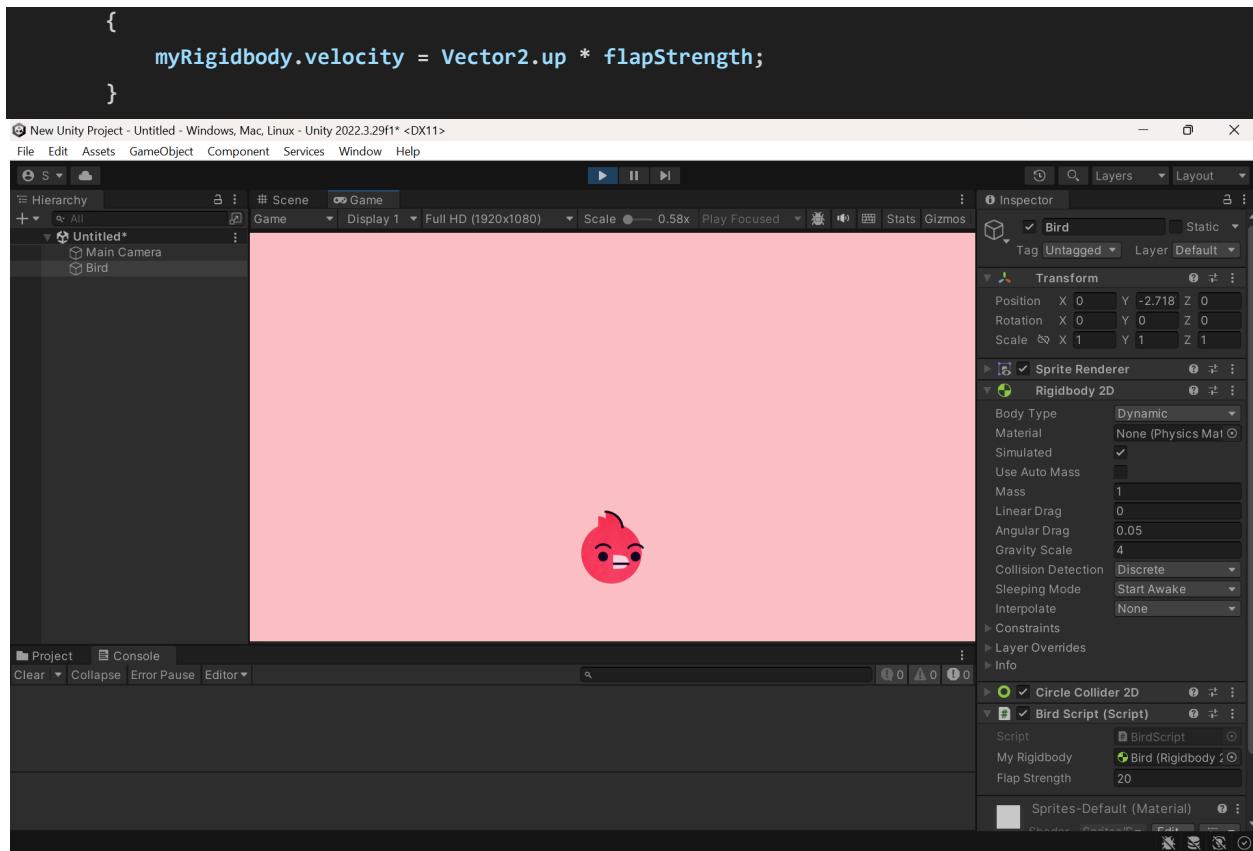
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space) == true)
        {
            myRigidbody.velocity = Vector2.up * 10;
        }
    }
}
```

4. Once the code is entered, we can click on the space bar to view our bird flap up and do.



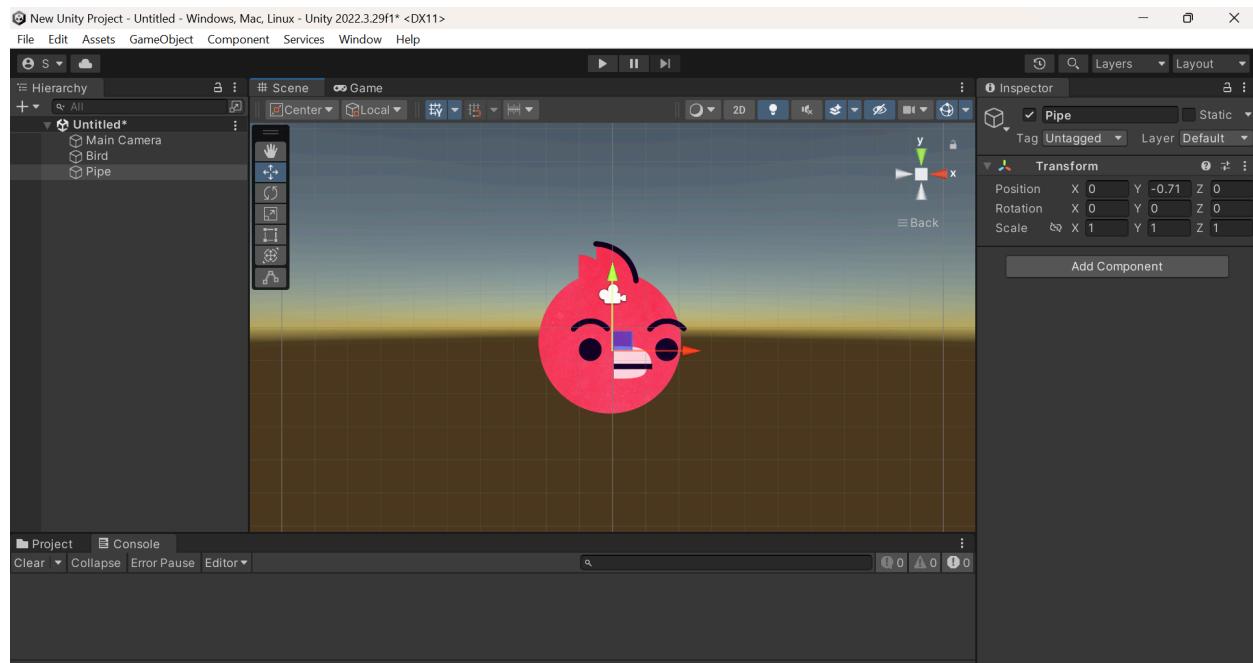
5. We then add flapStrength to our code to manipulate the up and down motion of the bird. We can also change the Gravity of the Rigid Body to make it more interactive.

```
public float flapStrength;
if (Input.GetKeyDown(KeyCode.Space) == true)
```

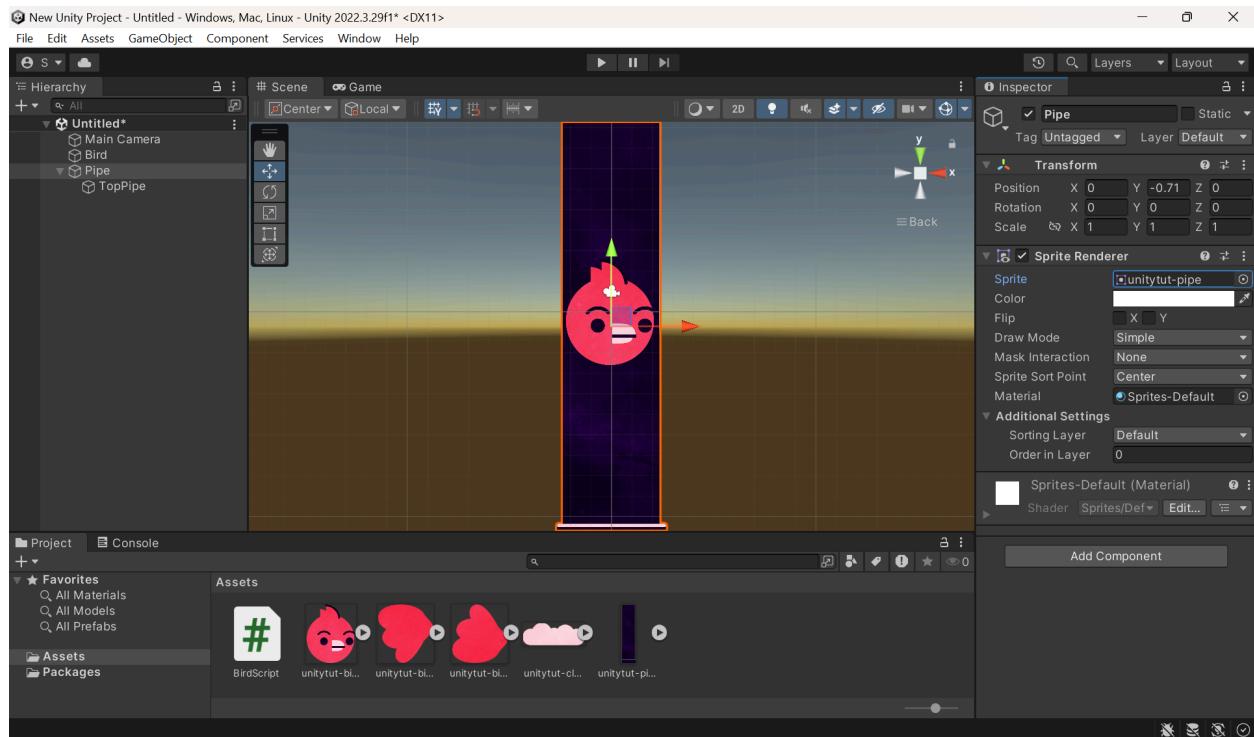


Step 3: Create spawning objects

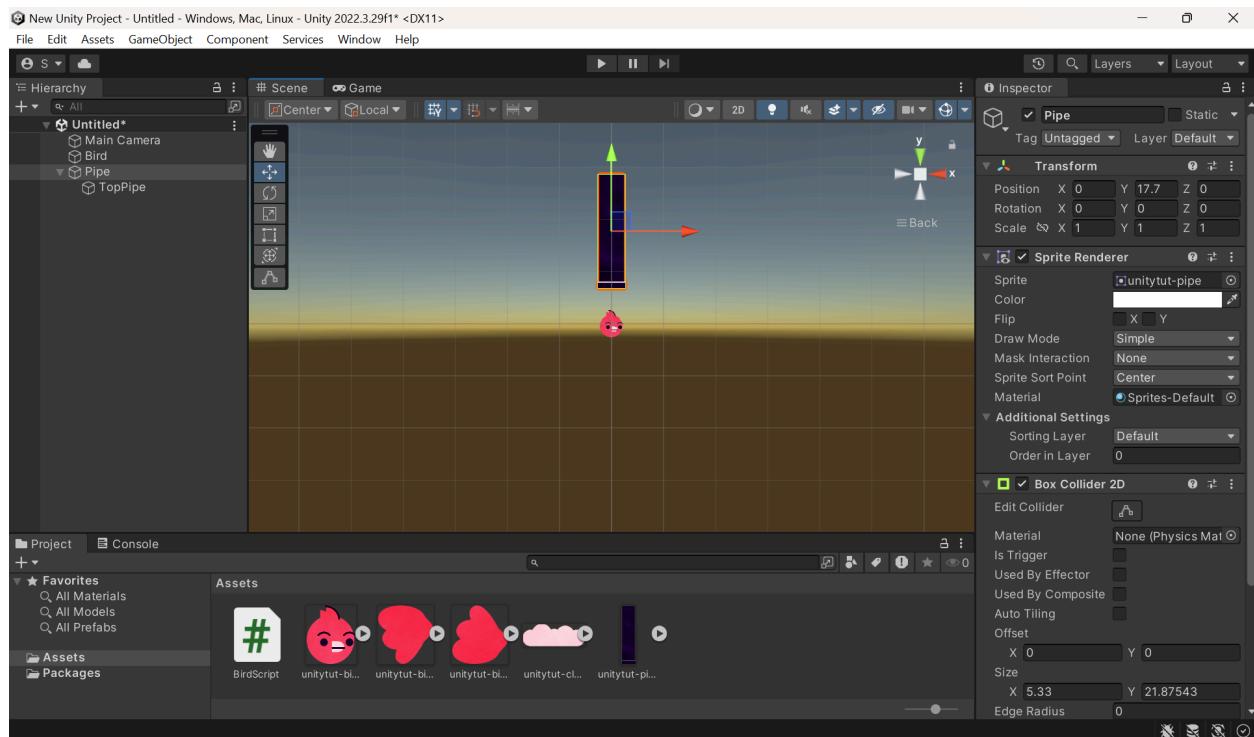
1. Create another GameObject => Rename it as Pipe



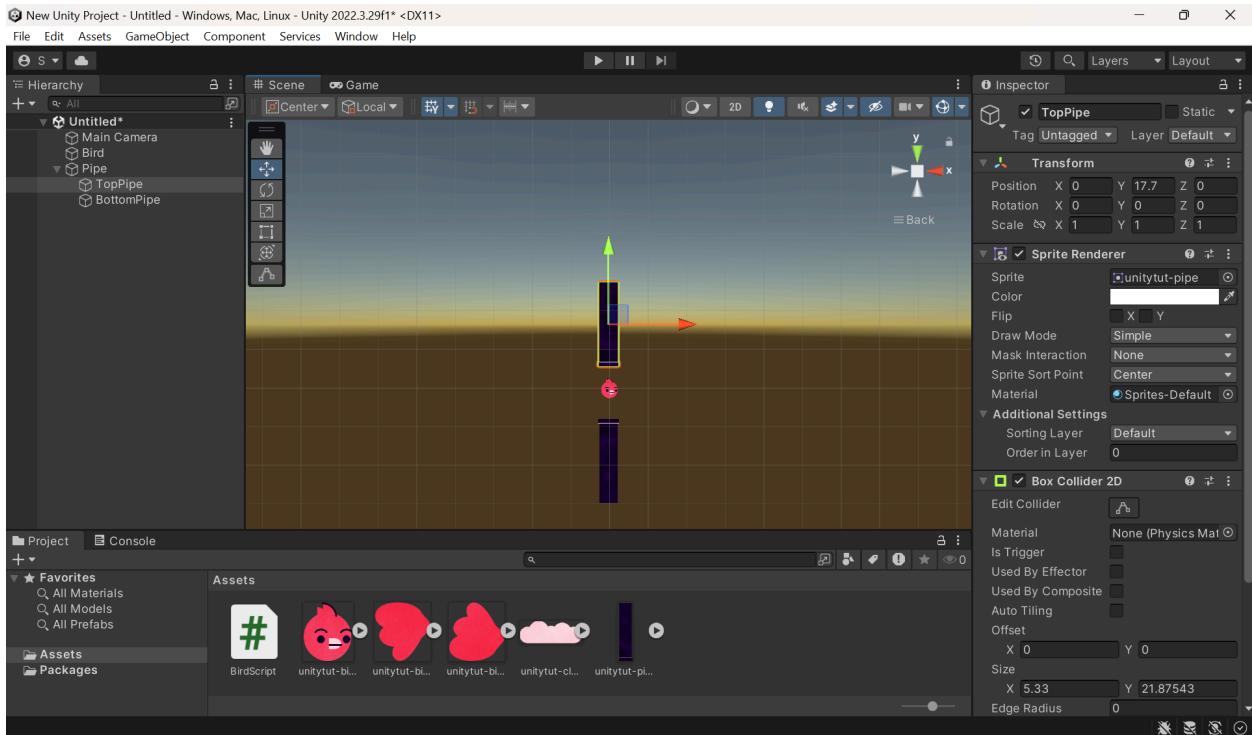
- Create another GameObject inside of Pipe => which is its child and name it TopPipe.
Then we add the Sprite Component to the Pipe object as we did it for the bird.



- Add a BoxCollider 2D, we don't need a Body for the pipe since it is not going to be affected by physics. Then we move the Pipe Object up above the Bird, keeping the X position 0.

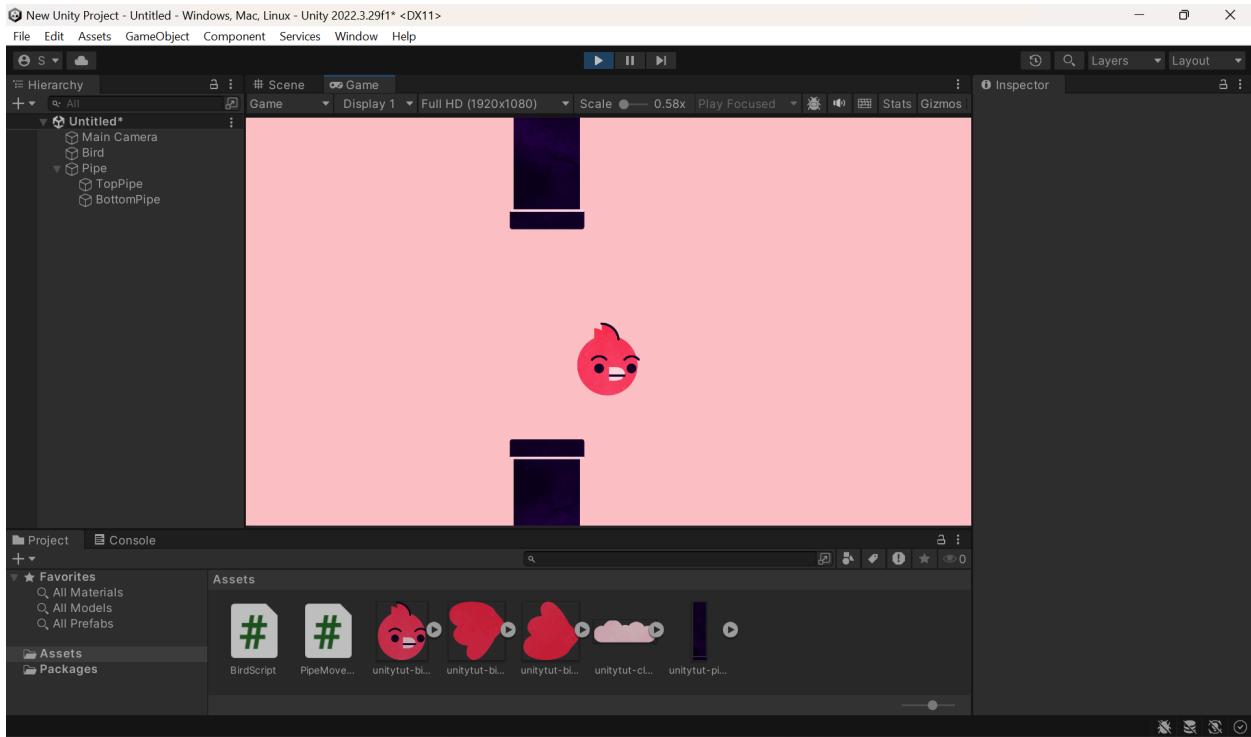


4. Duplicate the whole TopPipe Object and call it BottomPipe and turn it upside down by changing the Y scale and position to -1.

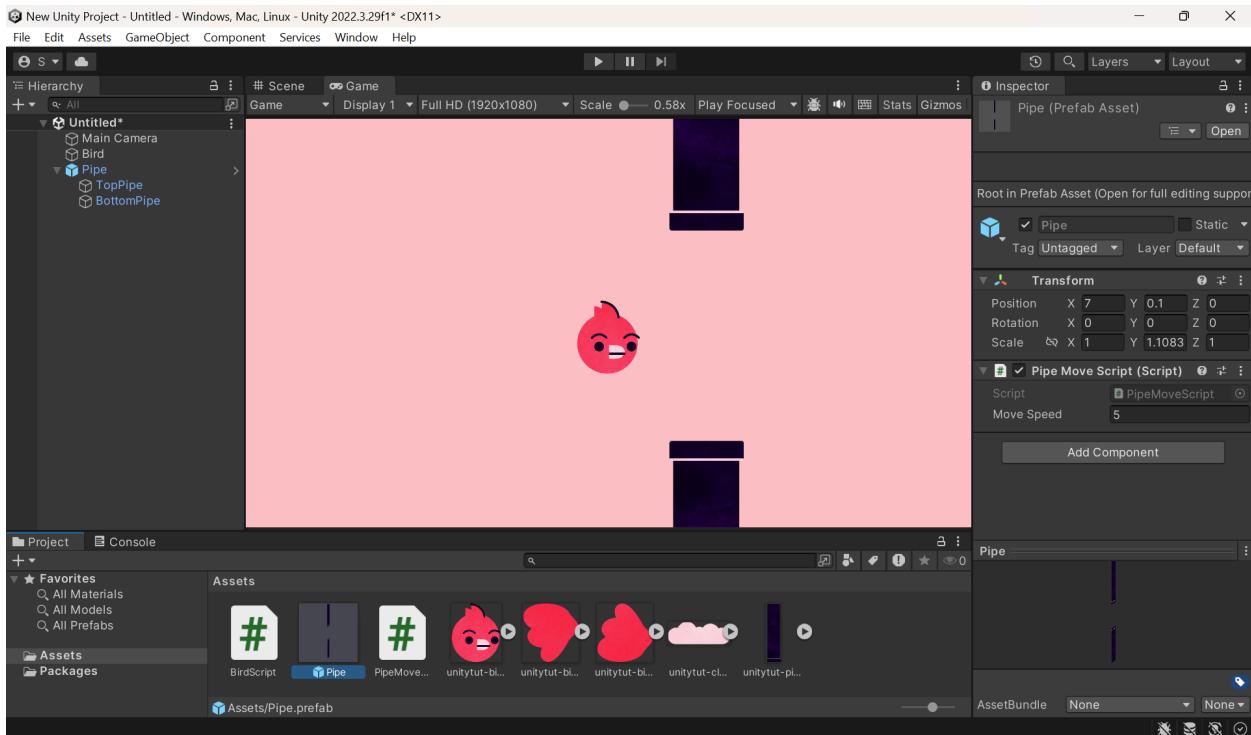


5. Now we create a New Script for the Pipe Object and name it PipeMoveScript, we give the following code in the script.

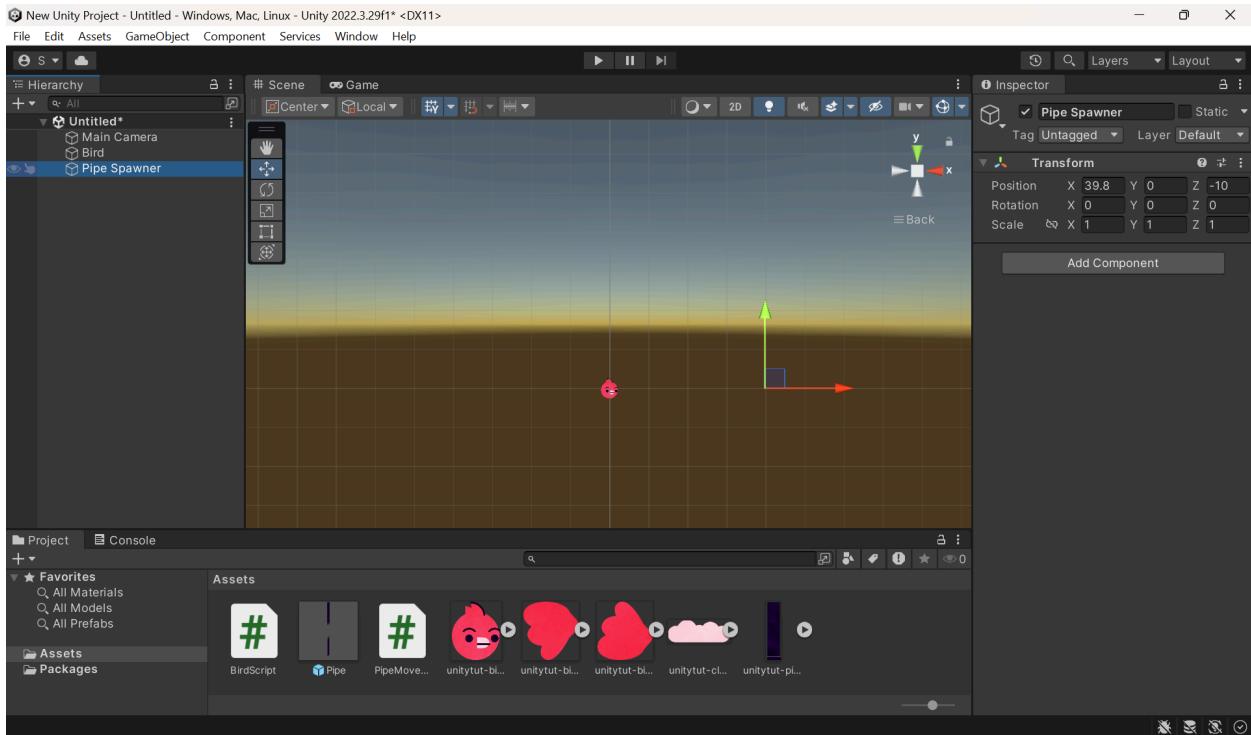
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class PipeMoveScript : MonoBehaviour
{
    public float moveSpeed = 5;
    void Start()
    {
    }
    void Update()
    {
        transform.position = transform.position + (Vector3.left * moveSpeed) * Time.deltaTime;
    }
}
```



6. Now create more spawning objects, for this drag the parent Pipe object into the Assets panel - Prefabricated GameObject - a blueprint of the GameObject.



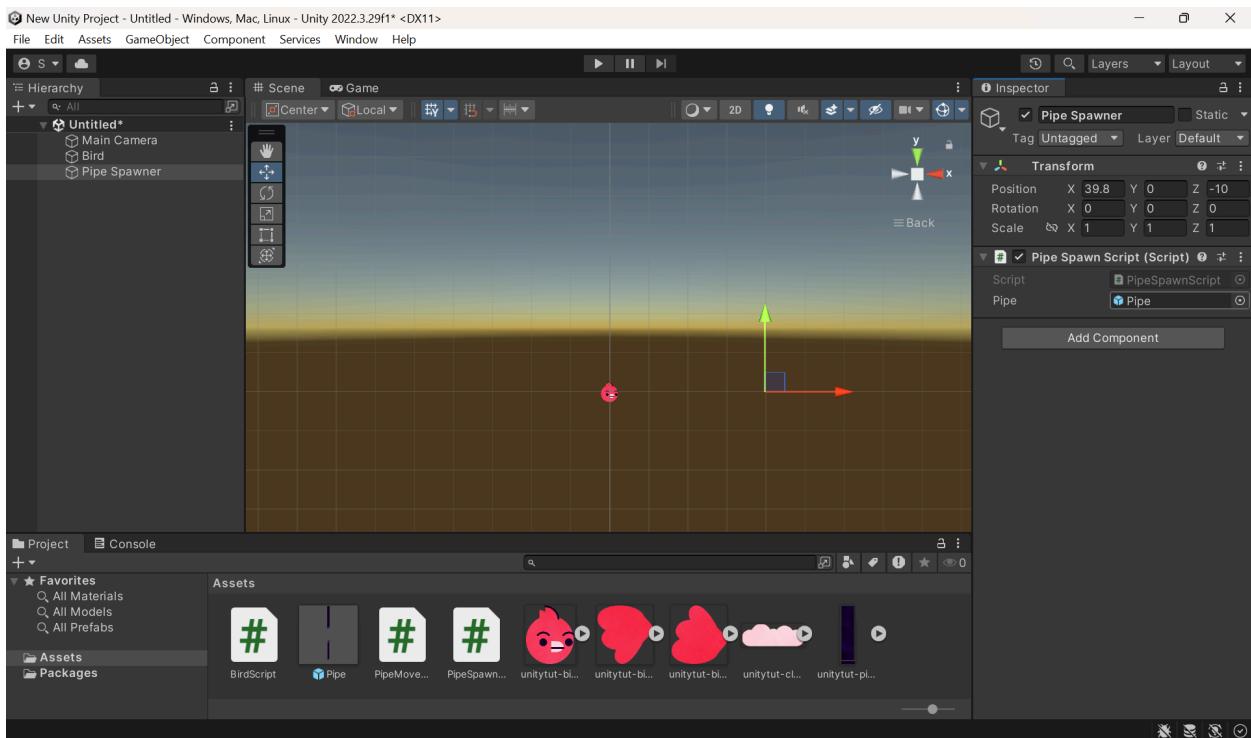
7. Now, we'll delete the Pipe Object and create a new GameObject and name it PipeSpawner and place it right in front of the camera.



8. Create a New Script for the Pipe Spawner Object, to create a new Pipe Object for every few seconds.

```
public GameObject pipe;
```

Add the Pipe Object as Component to the Pipe Spawner Object



9. Add the following code in C:\Users\sselc\Downloads\FlappyBird\New Unity Project\Assets\PipeSpawnScript.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class PipeSpawnScript : MonoBehaviour
{
    public GameObject pipePrefab;
    public float spawnRate = 2;
    private float timer = 0;
    public float heightOffset = 10;
    void Start()
    {
        spawnPipe();
    }
    void Update()
    {
        if (timer < spawnRate)
        {
            timer = timer + Time.deltaTime;
        }
        else
        {
            spawnPipe();
            timer = 0;
        }
    }
    void spawnPipe()
    {
        float lowestPoint = transform.position.y - heightOffset;
        float highestPoint = transform.position.y + heightOffset;
        Instantiate(pipePrefab, new Vector3(transform.position.x, Random.Range(lowestPoint, highestPoint), 0), transform.rotation);
    }
}
```

Add the following code in

C:\Users\sselc\Downloads\FlappyBird\New Unity Project\Assets\PipeMoveScript.cs

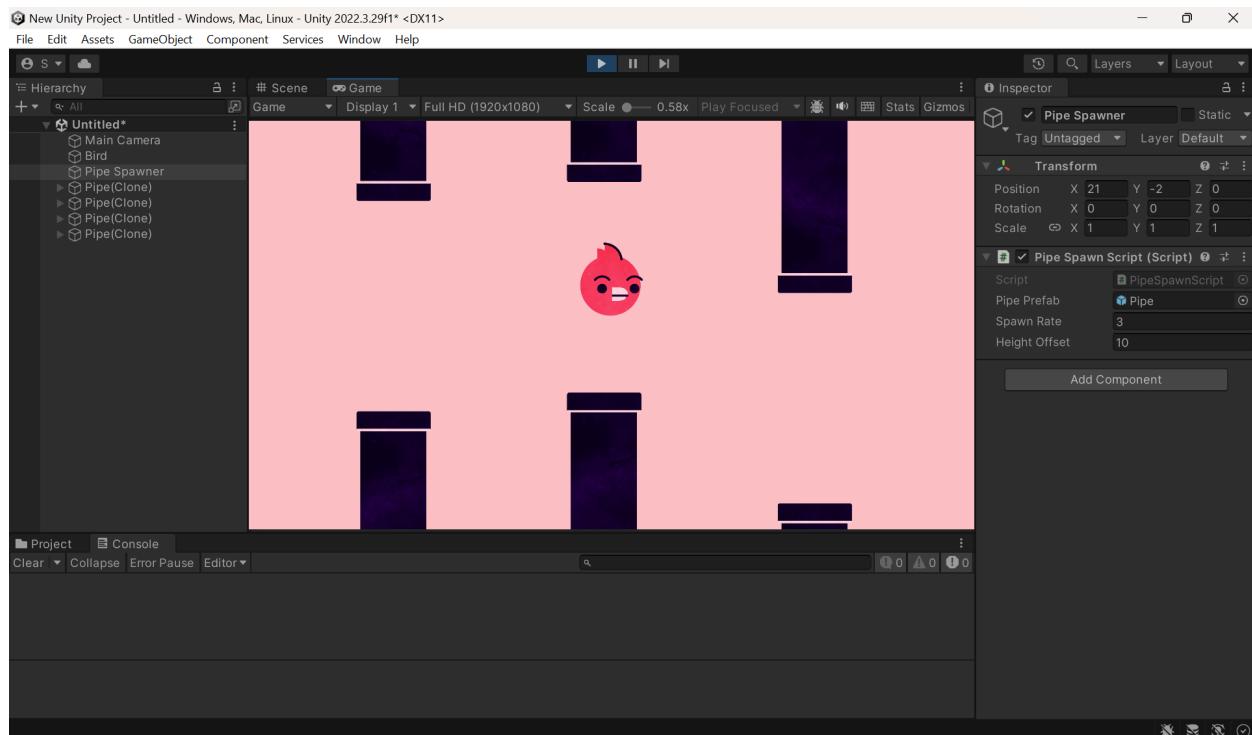
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class PipeMoveScript : MonoBehaviour
{
    public float moveSpeed = 5;
    public float deadZone = -45;
    void Start()
    {
```

```

        }

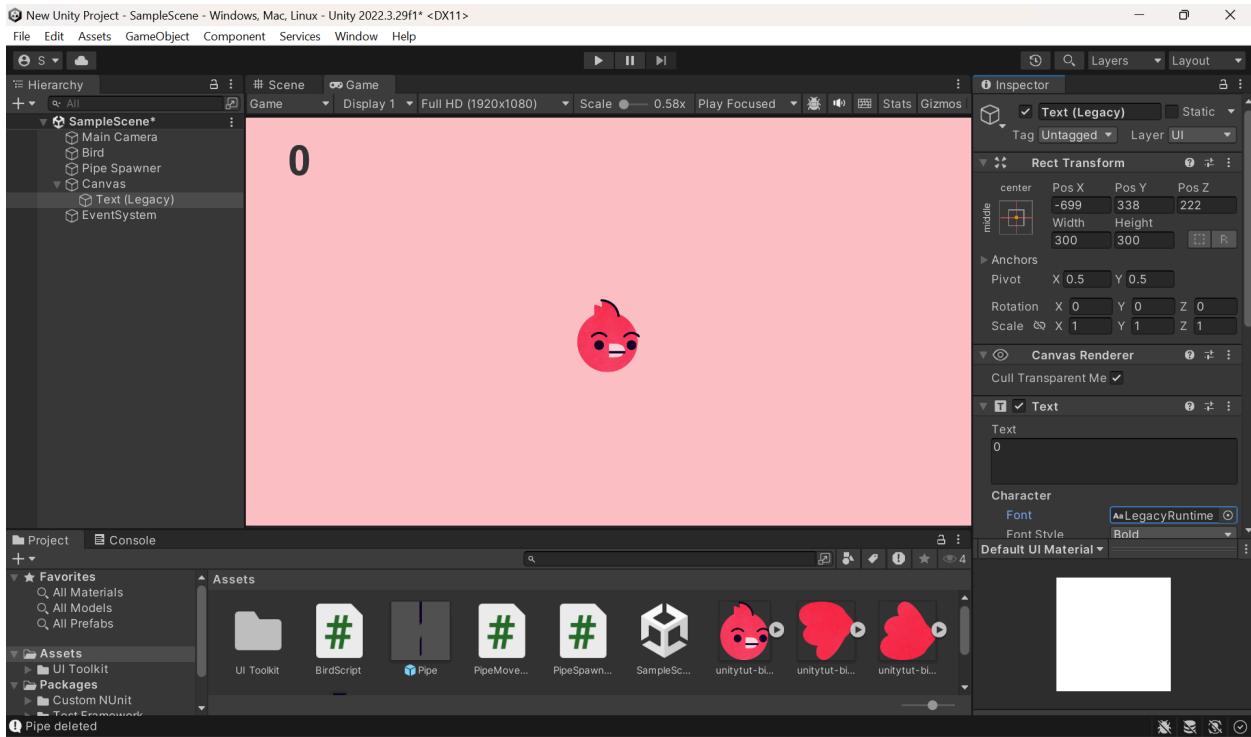
        void Update()
        {
            transform.position = transform.position + (Vector3.left * moveSpeed) * Time.deltaTime;
            if (transform.position.x < deadZone)
            {
                Debug.Log("Pipe deleted");
                Destroy(gameObject);
            }
        }
    }
}

```

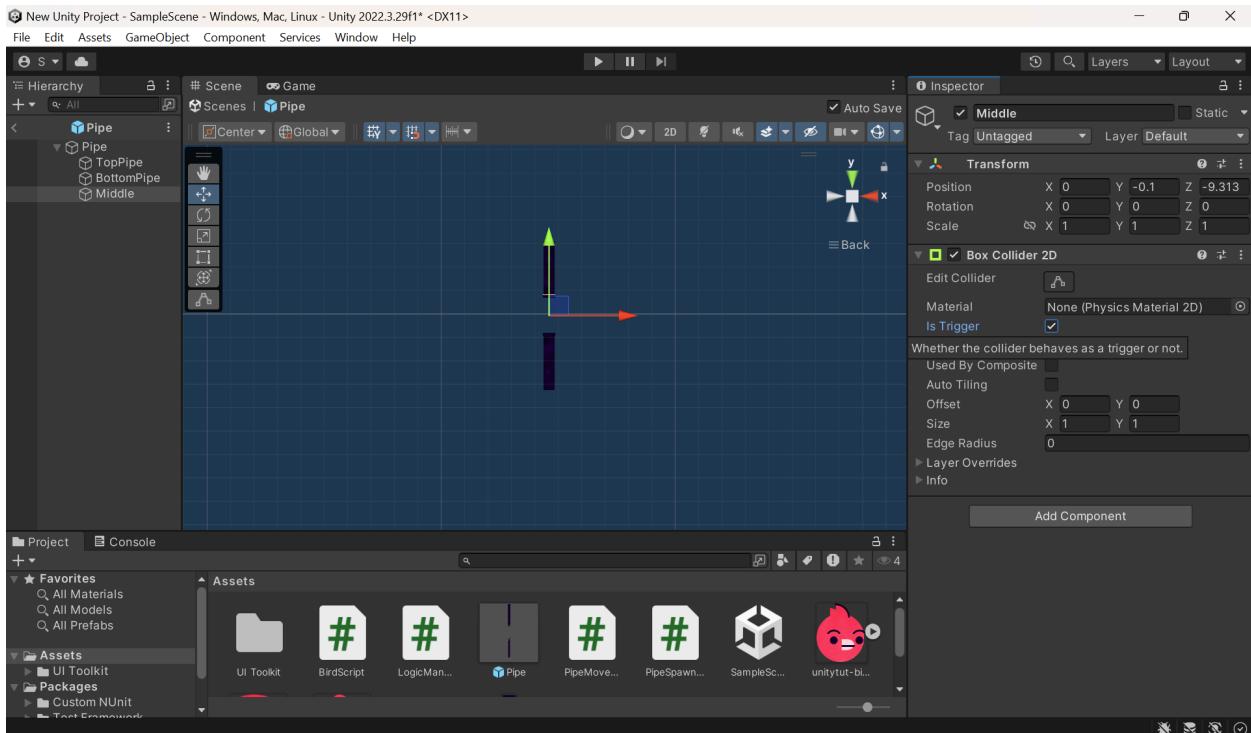


Step 4: Creating UI Elements

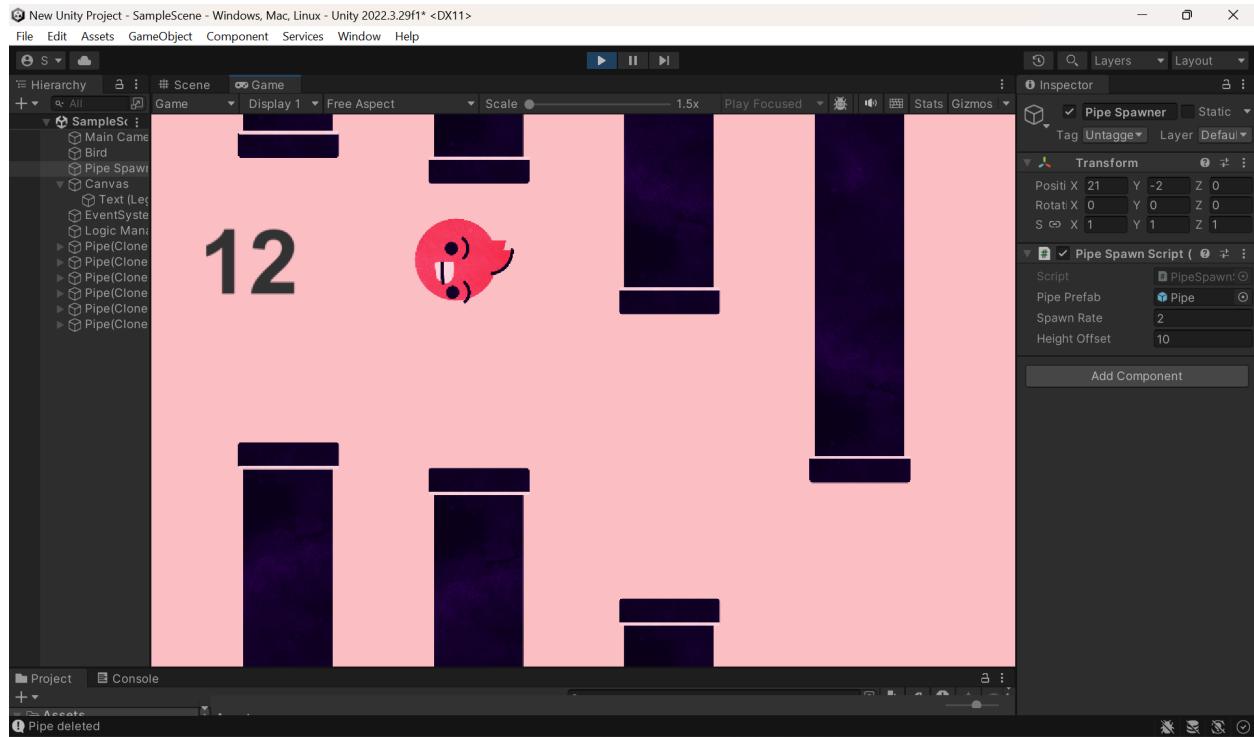
1. On the hierarchy panel, click left and select UI => Legacy => Text
2. To scale it evenly on all devices, we click on Canvas => UI Scale Mode => Scale with Screen Size
3. Change sensible reference resolution => X: 1920, Y: 1080
4. Change height, width, font size, etc



5. To keep track of the score, we create a new GameObject named Logic Manager => To keep track of the score, we add an invisible collider in between the Pipes, so that when the bird passes through the pipes the score increases.



6. After adding logic to the Logic Script, the score increases



COMPLETE SOURCE CODE:

[Assets\BirdScript.cs](#)

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

public class BirdScript : MonoBehaviour
{
    public Rigidbody2D myRigidbody;
    public float flapStrength;
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space) == true)
        {
            myRigidbody.velocity = Vector2.up * flapStrength;
        }
    }
}
```

Assets\LogicScript.cs

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

public class LogicScript : MonoBehaviour
{
    public int playerScore;
    public Text scoreText;
    [ContextMenu("Increase Score")]
    public void addScore(int scoreToAdd)
    {
        playerScore = playerScore + scoreToAdd;
        scoreText.text = playerScore.ToString();
    }

}
```

Assets\PipeMiddleScript.cs

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

public class PipeMiddleScript : MonoBehaviour
{
    public LogicScript logic;
    // Start is called before the first frame update
    void Start()
    {
        logic = GameObject.FindGameObjectWithTag("Logic").GetComponent<LogicScript>();
    }

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

    }
    private void OnTriggerEnter2D(Collider2D collision)
    {
        logic.addScore(1);
    }
}
```

Assets\PipeMoveScript.cs

```
using System.Collections;
```

```

using System.Collections.Generic;
using UnityEngine;

public class PipeMoveScript : MonoBehaviour
{
    public float moveSpeed = 5;
    public float deadZone = -45;
    // 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;
        if (transform.position.x < deadZone)
        {
            Debug.Log("Pipe deleted");
            Destroy(gameObject);
        }
    }
}

```

[Assets\PipeSpawnScript.cs](#)

```

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

public class PipeSpawnScript : MonoBehaviour
{
    public GameObject pipePrefab;
    public float spawnRate = 2;
    private float timer = 0;
    public float heightOffset = 10;
    void Start()
    {
        spawnPipe();
    }

    // Update is called once per frame
    void Update()
    {
        if (timer < spawnRate)
        {
            timer = timer + Time.deltaTime;
        }
    }
}

```

```
        }
    else
    {
        spawnPipe();
        timer = 0;
    }
}
void spawnPipe()
{
    float lowestPoint = transform.position.y - heightOffset;
    float highestPoint = transform.position.y + heightOffset;

    Instantiate(pipePrefab, new Vector3(transform.position.x, Random.Range(lowestPoint,
highestPoint), 0), transform.rotation);
}
}
```

CONCLUSION

The completion of this project results in a fully functional Flappy Bird game clone, demonstrating a comprehensive understanding of game development principles in Unity.