Part 3: Unity Flappy Bird Tutorial

Contents

Part 3: Unity Flappy Bird Tutorial 1

The Pipes 1

Obstacle Spawner 4

Game Play Settings 7

Assignment Submission...8

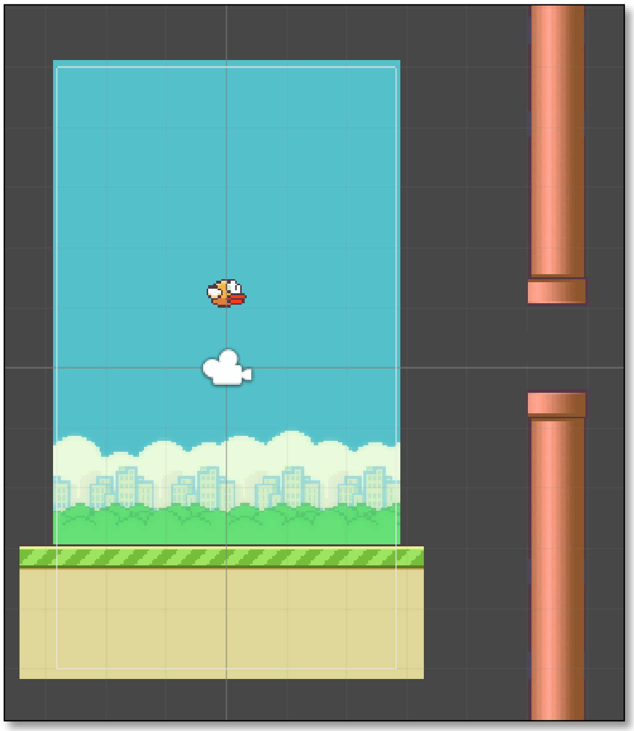
The Pipes

Time for us to create the obstacles (commonly known as the warp pipes).

NOTE: There are 2 different colored pipes, it doesn't matter which ones you pick.

1. **Assets** → **Sprites** → Drag 2 pipes one at a time onto the **Hierarchy**.
2. Use the **Inspector** to name one pipe **PipeTop**,
   a. **Sprite Renderer** → **Order in Layer**: **1**
3. Use the **Inspector** to name the other pipe **PipeBottom.**
   a. **Sprite Renderer** → **Order in Layer**: **1**
4. Drag the pipes to the right of the scene as shown below.

NOTE: It is important that the pipes are aligned vertically for proper game play. The vertical and horizontal distance can be adjusted to your preference whenever you wish. If the pipes are too close, your bird won't have a chance to start flying before being run over by a pipe.



Copper Pipes Only: If you picked the Copper pipes, **PipeTop** needs to be rotated 180 degrees.

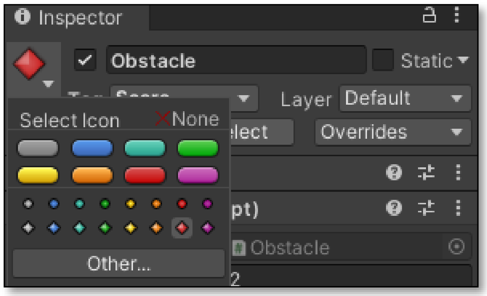
Select **PipeTop** → **Inspector** → **Rotation X**: 180

5. Select both pipes → **Inspector** → **Add Component** → **Box Collider 2D**.

6. **Hierarchy** → **Right Click** → **Create Empty** This will create an empty GameObject.

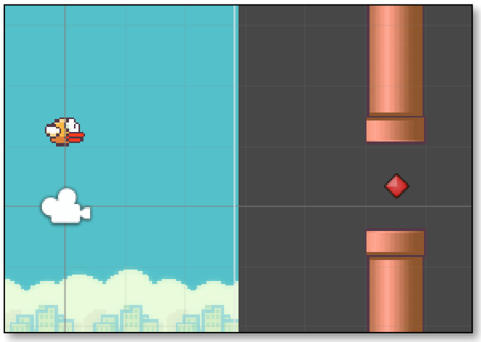
7. **Inspector** → name it **Obstacle**

8. **Obstacle** is hard to see on the Scene. Add an icon as shown below. The icon is only for your reference, it will not show up when the game is being played.

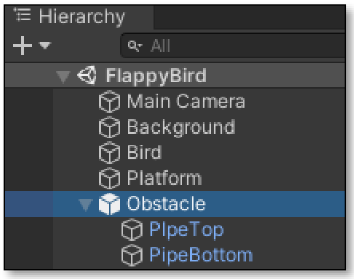


9. Select the **Move** tool.

10. Use the **Move** tool to move the **Obstacle** to the right between the pipes as shown below.



11. **Hierarchy** → Drag the two pipes into **Obstacle** as shown below.



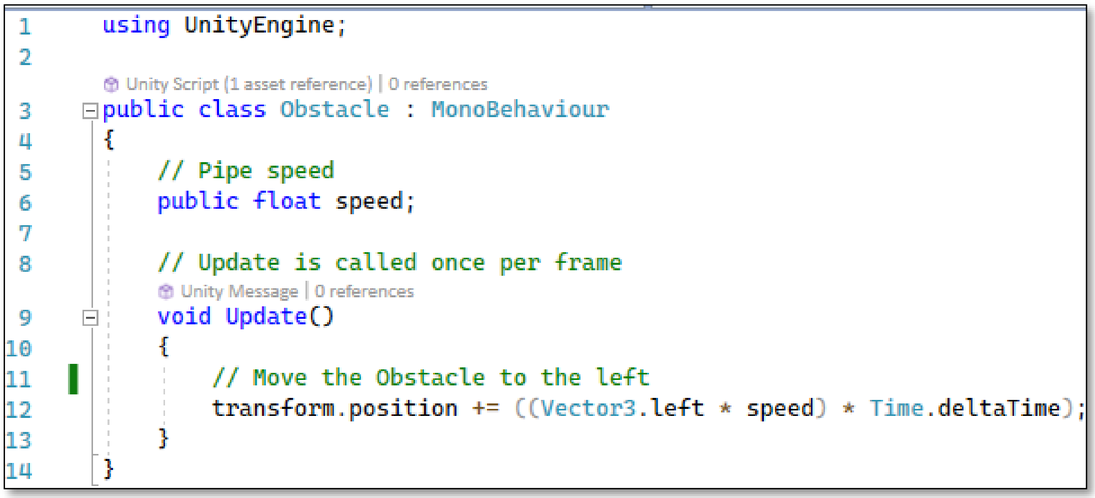
The Obstacle game object is now the parent of the pipes.

12. Create a new C# script named **ObstacleScript**

13. Double-click **ObstacleScript** to open Visual Studio Community 2022.

14. **Add** and **save** the following code:

NOTE: Delete the Start() function as it is not needed.



```
using UnityEngine;

Unity Script (1 asset reference) | 0 references
public class Obstacle : MonoBehaviour
{
    // Pipe speed
    public float speed;

    // Update is called once per frame
    Unity Message | 0 references
    void Update()
    {
        // Move the Obstacle to the left
        transform.position += ((Vector3.left * speed) * Time.deltaTime);
    }
}
```

15. Save the script.

16. Go back to the **Unity Editor**.

17. In the **Hierarchy** → Drag the **ObstacleScript** script to the **Obstacle** game object.

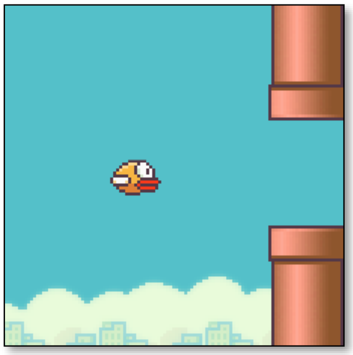
18. In the **Hierarchy** → Select the **Obstacle** game object.

19. In the **Inspector** → **Obstacle (Script)**

    a. **Speed:** 2

20. Click **Play** to test the game. A single set of pipes should move across the scene as shown. You can fly your bird between the pipes.

21. Click **Play** again to stop, click **Play** to start again.

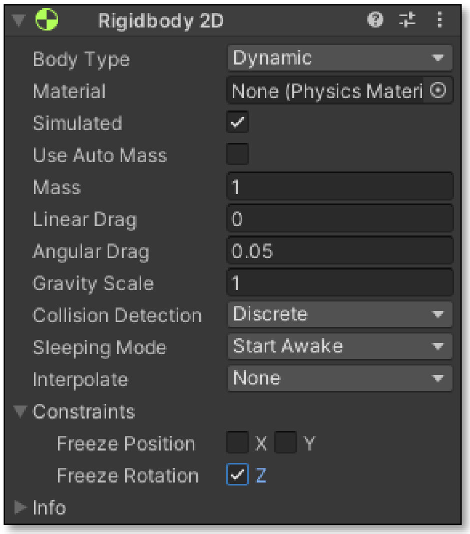


22. Notice that if your bird hits the pipe just right, it flips over. Let's fix that.

23. Select the **Bird** game object in the **Hierarchy**

24. **Inspector** → **Rigidbody 2D** → **Constraints**

25. Enable **Freeze Rotation Z** as shown on the right.



Rigidbody 2D

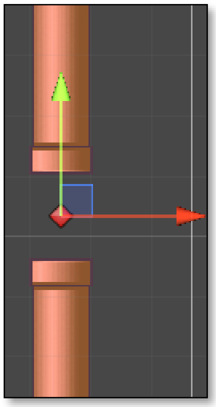
Body Type	Dynamic
Material	None (Physics Materi
Simulated	✓
Use Auto Mass	□
Mass	1
Linear Drag	0
Angular Drag	0.05
Gravity Scale	1
Collision Detection	Discrete
Sleeping Mode	Start Awake
Interpolate	None
Constraints	
Freeze Position	□ X □ Y
Freeze Rotation	☑ Z
Info	

Your Flappy Bird stays upright as it flies. This is starting to look like a real game! We need more pipes . . .

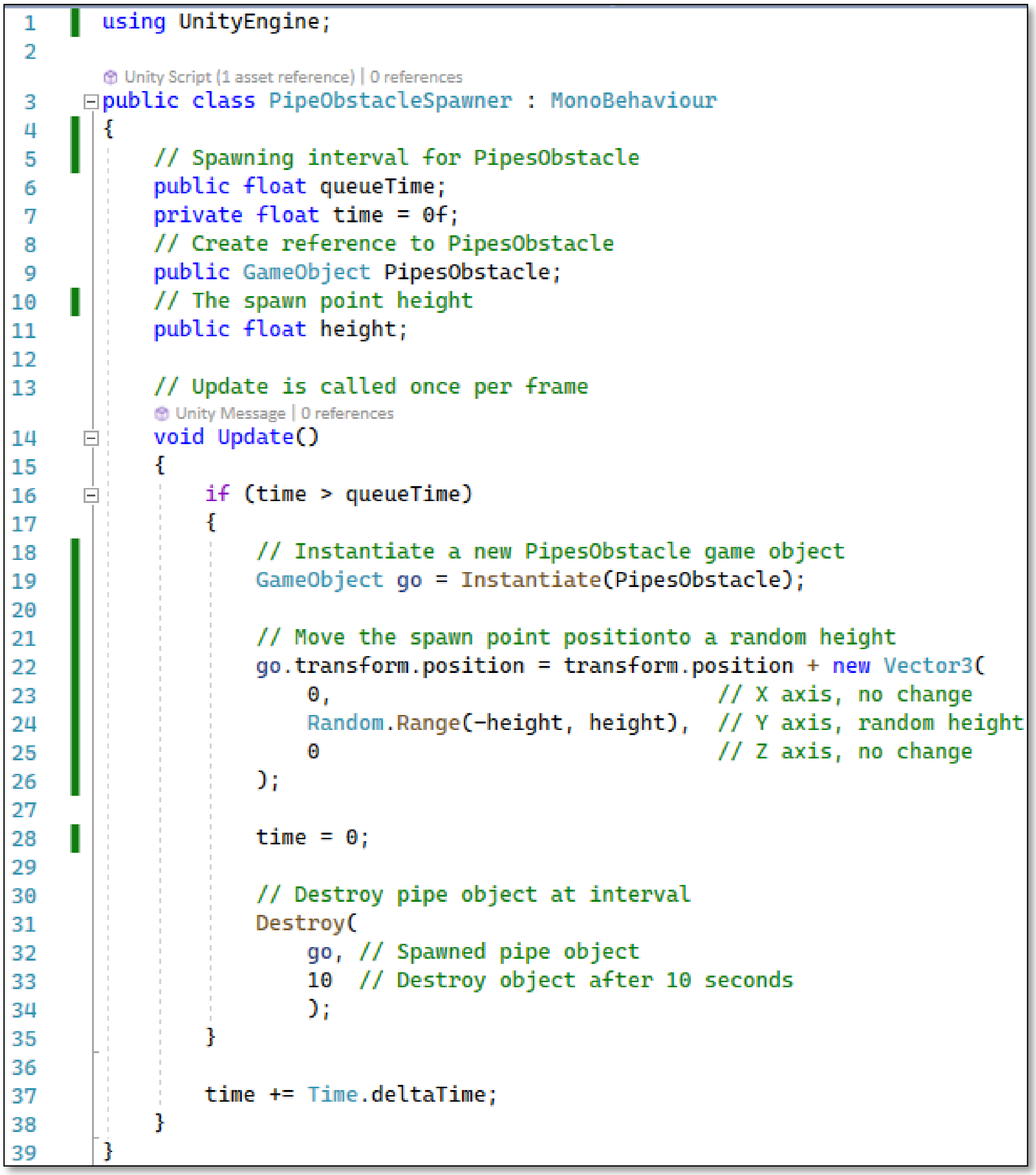
Obstacle Spawner

Our obstacle is ready. Let's create an obstacle spawner which will give us an endless supply of pipes to avoid.

What is spawning? In video games, spawning is the live creation of an entity. A spawn point is where the entity is spawned.



1. **Hierarchy** → **Right-click** → **Create Empty** → name it **ObstacleSpawner**
2. **Inspector** → Choose a different color icon
3. Click the **Move** tool. This will show the location of the **ObstacleSpawner**, which is the spawn point.
4. In the image to the right, the **ObstacleSpawner** is right on top of the **Obstacle**. When new pipes are created, they will start from the same horizontal distance as the first set of pipes.
5. If your **ObstacleSpawner** isn't located there, move it on top of the **Obstacle**.
6. In the **Scripts** folder, → Create a C# script named **ObstacleSpawnerScript**
7. Double Click to edit.
8. Add the following code.



```
using UnityEngine;

Unity Script (1 asset reference) | 0 references
public class PipeObstacleSpawner : MonoBehaviour
{
    // Spawning interval for PipesObstacle
    public float queueTime;
    private float time = 0f;
    // Create reference to PipesObstacle
    public GameObject PipesObstacle;
    // The spawn point height
    public float height;

    // Update is called once per frame
    Unity Message | 0 references
    void Update()
    {
        if (time > queueTime)
        {
            // Instantiate a new PipesObstacle game object
            GameObject go = Instantiate(PipesObstacle);

            // Move the spawn point positionto a random height
            go.transform.position = transform.position + new Vector3(
                0,                              // X axis, no change
                Random.Range(-height, height),  // Y axis, random height
                0                               // Z axis, no change
            );

            time = 0;

            // Destroy pipe object at interval
            Destroy(
                go, // Spawned pipe object
                10  // Destroy object after 10 seconds
                );
        }

        time += Time.deltaTime;
    }
}
```

1. Save the script.

2. Drag the **ObstacleSpawnerScript** script to the **ObstacleSpawner** game object.

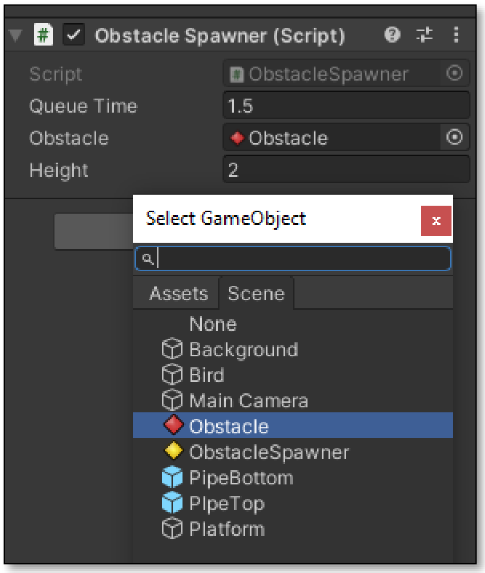
3. In the **Hierarchy** → Select the **ObstacleSpawner** game object.

4. In the **Inspector** → **Obstacle Spawner (Script)**

    a. **Queue Time:** 1.5

    b. **Obstacle:** Select **Obstacle** as shown below

c. **Height:** 2



Obstacle Spawner (Script)

Script: ObstacleSpawner

Queue Time: 1.5

Obstacle: Obstacle

Height: 2

Give it a try. You should have an endless supply of Pipes to fly between as shown below.

Game Play Settings

In the **Inspector** → You can adjust the **Obstacle (Script)** and **ObstacleSpawner (Script)** settings to change the behavior of the game play.

Bird script

- **Velocity:** How far the bird can fly (jump)
  Initial setting: 3

Obstacle script

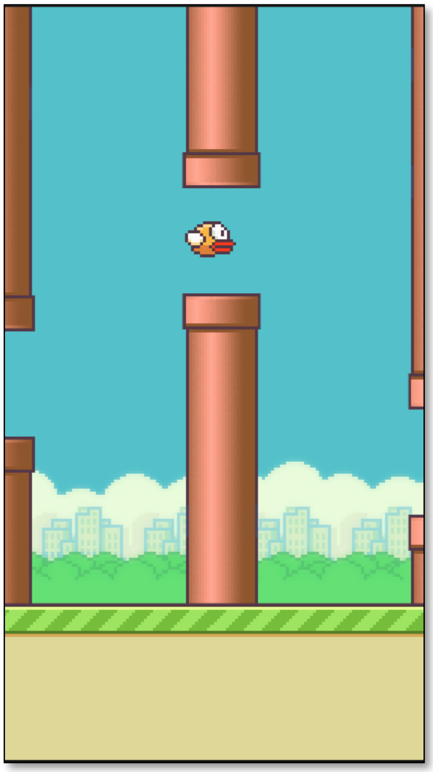
- **Speed**: How fast the pipes move from the right to the left.
  Initial setting: 2

ObstacleSpawner script

- **Queue Time:** Time between spawning a new set of pipes.
  Initial setting: 1.5 seconds

- **Height:** How high and low the pipes randomly spawn. A lower number will spawn closer together, higher would be further apart.
  Initial setting: 2

Example run:



Assignment Submission

A Unity project is at least 200 MB. That is too big to be submitted.

Please attach a screenshot of your project to the assignment in Blackboard.