Component: Pipe

Student Number: C00157047
Student Name: Martin Farrell

Introduction

A pipe is an object that allows another object to pass through it and out the other side.

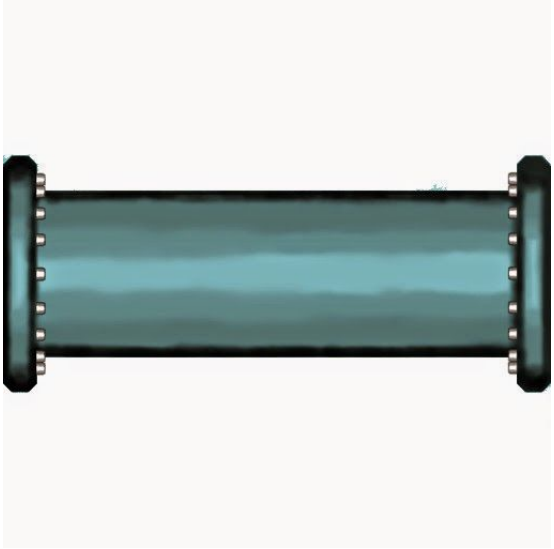


Fig.1 a simple pipe

Usage

To add the pipe to a Unity project as a git submodule, perform the following steps:

1. Download project from my repository.
2. Inside the prefabs folder right-click the pipe prefab and click export package.
3. Ensure that the “pipe.cs”, pipe sprite and pipe sound are ticked before exporting(Not having an audio clip will cause the scene not to compile, you can just add your own source mp3 or wav file to resolve this).
4. Using the file explorer drop the exported package into the assets folder of your desired scene.
5. Click the exported package in the editor and drag the generated prefab into your scene. If you chose to use my sound effect it should work. If not you will have to drag and drop your own into the audio source.

Once you have added the pipe, you can use it as follows:

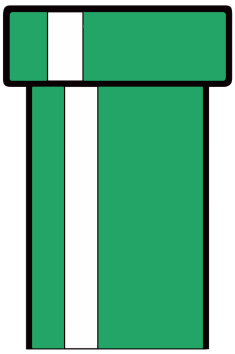
1. Just drag it anywhere on your scene.

2. Any object with simulated physics that collides with the top or bottom will teleport to the other side.

Features

Feature #1: Basic Pipe Functionality downwards [20 points]

The basic pipe functionality in the downward vertical direction.



Conditions of satisfaction:

- The pipe can be assigned a fixed position.
- When an object collides with the top of the pipe it teleports to the bottom of the pipe.

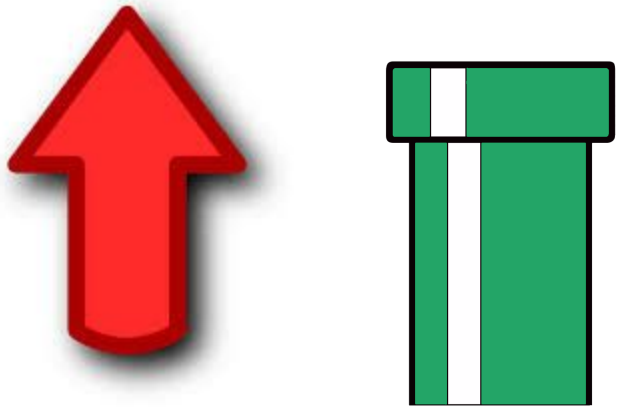
Example: If a ball were to fall under the effects of gravity and make direct contact with the top of the pipe it will teleport to the other side as if it had gone “through” the pipe. This is done using onCollisionEnter with this line of code:

```
otherObj.SetPositionAndRotation(new Vector3(otherObj.position.x, otherObj.position.y - this.GetComponent<Renderer>().bounds.size.y - otherObj.GetComponent<Renderer>().bounds.size.y - 0.05f, otherObj.position.z), new Quaternion());
```

Source Code: Source code
Test Code:No tests were written
Video Capture: No tests were written
JIRA: No Jira link

Feature #2: Pipe Functionality upwards [20 points]

Same functionality as #1 but in an upward direction.



Conditions of satisfaction:

- When an object collides with the bottom of the pipe it teleports to the top of the pipe.
- The object will not teleport back and forth between the top and bottom of the pipe.

Example: If a ball were travelling with an upwards velocity directly beneath the pipe and made direct contact with the bottom of the pipe, it will teleport to the other side as if it had gone “through” the pipe. This is done using onCollisionEnter with this line to check if the object is beneath:

collision.gameObject.GetComponent<Rigidbody2D>().transform.position.y < this.GetComponent<Transform>().position.y

Source Code:Source code
Test Code: No tests were written
Video Capture: no tests were written
JIRA: no Jira link

Feature #3: Sound Effects [10 points]

- An object falling from above and colliding with the pipe will create a sound effect.
- An object moving upwards which collides with the bottom of the pipe will also cause a sound effect

If an object were to make direct contact with the top or bottom of the pipe, it would play the desired sound effect. This is done using onCollisionEnter with this line of code provided you have loaded the file of your choosing:

pipeSource.Play();

Source Code:Source code
Test Code: No tests were written
Video Capture: no tests were written
JIRA: no Jira link

Feature #4: On/off capabilities [15 points]

In case anyone wishes to turn a pipe on or off programmatically for whatever reason the user will be able to do so by simply calling a function.

Conditions of satisfaction:

- If “active” pipe will become inactive.
- If inactive, pipe will become active.

By calling the function “changeAlive()” the user will get access to this code:

```
if(active)
    {
       active = false;
    }
    else
    {
       active = true;
    }
```

Which will function like a switch of sorts.

Source Code:Source code
Test Code: No tests were written
Video Capture: no tests were written
JIRA: no Jira link

Feature #5: Animation [15 points]

To give the pipe a bit more life I’m going to add an idling animation. The pipe will rotate very slightly left and right before returning to the centre.

Conditions of satisfaction:

- The animation plays correctly.
- The animation loops continuously.
- The pipe returns to centre.

The animation was completed in Unity’s UI and there were no programming aspects.

Source Code: No code involved
Test Code: No tests were written
Video Capture: no tests were written
JIRA: no Jira link