## horizontal line

Functioning Collision System

13.01.2021

**─**

Lewis McIntyre

### S2028246

Computer Games Software Development Year 1

# Overview

I *confirm that the code contained in this file (other than that provided or authorised) is all my own work and has not been submitted elsewhere in fulfilment of this or any other award*.

*Signature*.

# Contents

Page 1 - Contents

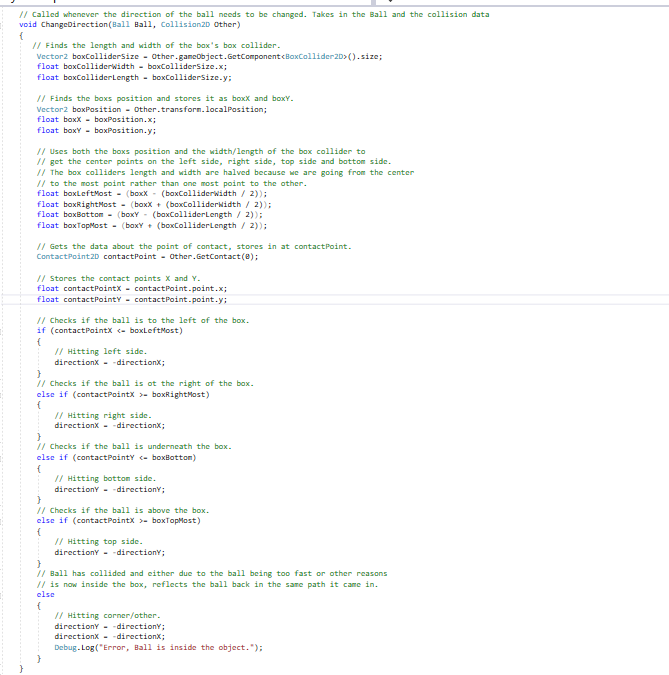
Page 2 - Code

1. The change direction function code.
2. Where the function is called code.

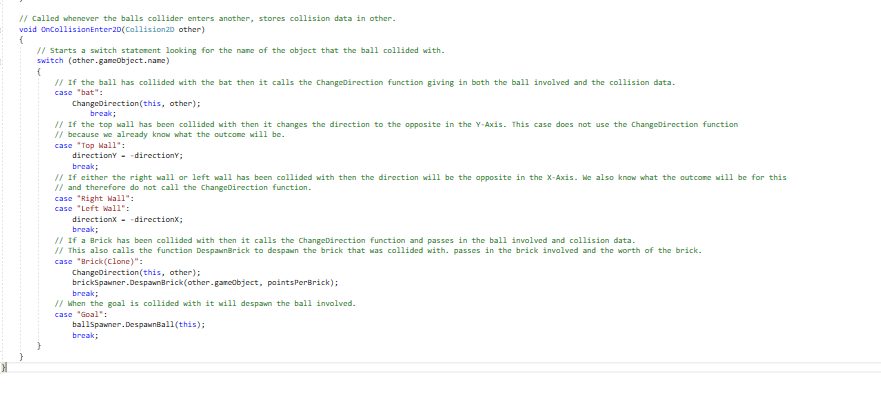
Page 3 - Explanation

1. Code 1
   1. Variables and terms
   2. All the object data used
   3. How the balls are reflected using the object data
2. Code 2
   1. Description of how the function is called.
3. Examples

**Code**

- Code 1

Code 2-

****

**Explanation**

**Code 1**

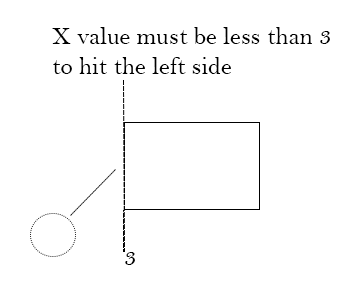
The object that has been collided with is called box in the code because the code can only work if the object has a BoxCollider2D element. The contact point refers to the points X and Y at which the ball collided with the box.

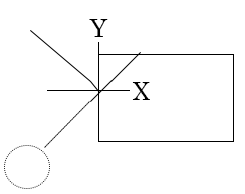
When the ChangeDirection function is called, it takes in the Ball and the collision data. The dimensions of the ball are found using the box collider's size and the position uses the box’s local position at the time of the collision. Using these we can calculate the top-most point, bottom-most point, left-most point and right-most point. We do this by adding or removing half the box’s width and length depending on which side we are looking for. For example, when getting the left-most we want to take away half the box colliders width, but if we wanted the right-most, we would add half the box colliders width. We are also only adding half of the width and length because transform.localPosition gives the center point of the object and to get to the left point or the right point we only need half of the distance.

After all the sides of the box are found, the only thing left to find is the point of contact. Using the first object that the ball collided with we find the point of it on the x and y axis. With both the contact point and the sides of the box we can now find which side it is on and properly reflect the ball off the box. The contact points x value is checked against the box’s left and right side. If the box has a larger x value than the box’s right-most side, then the ball has collided with the right side. If the box has a lower x value than the box’s left-most side, then the ball has collided with the left side. **(See Example 1)**. The same is repeated for the top and the bottom with the y value. With these we can then treat them exactly how we treat the walls and just flip either the x or the y direction value depending on what side is hit. For example, if the left side is hit then we have to flip the x direction value. **(See Example 2).**

**Code 2**

Shows how the ChangeDirection function is called. It is not called when hitting the top wall, right wall or left wall because we already know which direction it will bounce.

**(Ex.1)**

**(Ex.2) Represents X-Axis being flipped.**