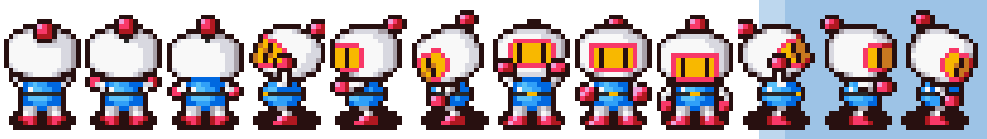
Interactive Media Development



Daniel Mateus Pires,

X00132886

Institute of Technology

Tallaght

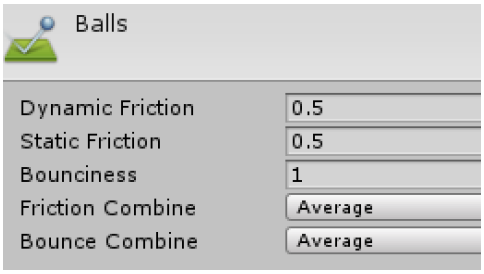
TABLE OF CONTENTS

GAME FEATURES

COLLISIONS BETWEEN BALLS, TABLE AND EDGES

Physics had to me implemented for the different materials. The collisions with the Cue and the other elements needed logic to be deactivated / reactivated on actions.

Example of material:



PLAYER CHOOSES DIRECTION OF CUE

Complex angle calculation had to be done

```
cueCollider.enabled = false;

Vector3 mousePos = Input.mousePosition;
Vector3 diffMousePos = mousePos - lastMousePos;
lastMousePos = mousePos;

if (diffMousePos.x > 0)
    transform.Translate(Vector3.right * 0.1f * diffMousePos.x);
else
    transform.Translate(Vector3.left * 0.1f * -diffMousePos.x);



transform.LookAt(whiteball.transform);

transform.position = (transform.position - whiteball.transform.position).normalized * 10 + whiteball.transform.position;

if (Input.GetMouseButtonDown(0))
{ // left click
    Vector3 shoot = 1000 * transform.forward;
    cueRigidBody.AddForce(shoot);
    shooting = true;
}
```

CUE TARGETS WHITE BALL

See above code, white ball is found and targeted

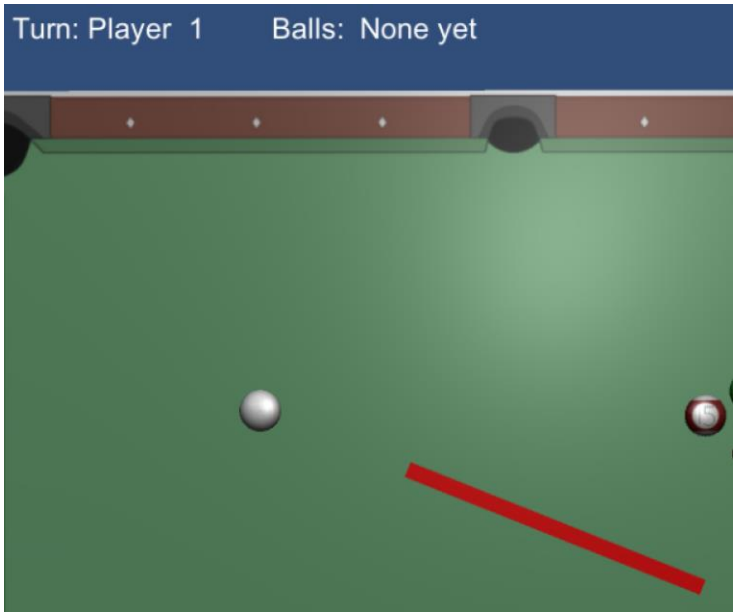
GAME RESETS IF BLACK BALL POTTED BEFORE OTHER COLORED BALLS

```
if (countColoredBalls < 7)
    SceneManager.LoadScene("Demo");
```

Scene is reloaded

MULTIPLAYER

Color of Cue and UI changes when player turn ends.



KNOWS WHAT COLOR OF BALLS IS ASSIGNED TO EACH PLAYER

CHALLENGES

It was a very challenging assignment overall, the time constraint and the very low preparation on the framework made the project very high time consuming and frustrating.

The biggest challenge was to implement the mathematics behind the rotation of the cue.