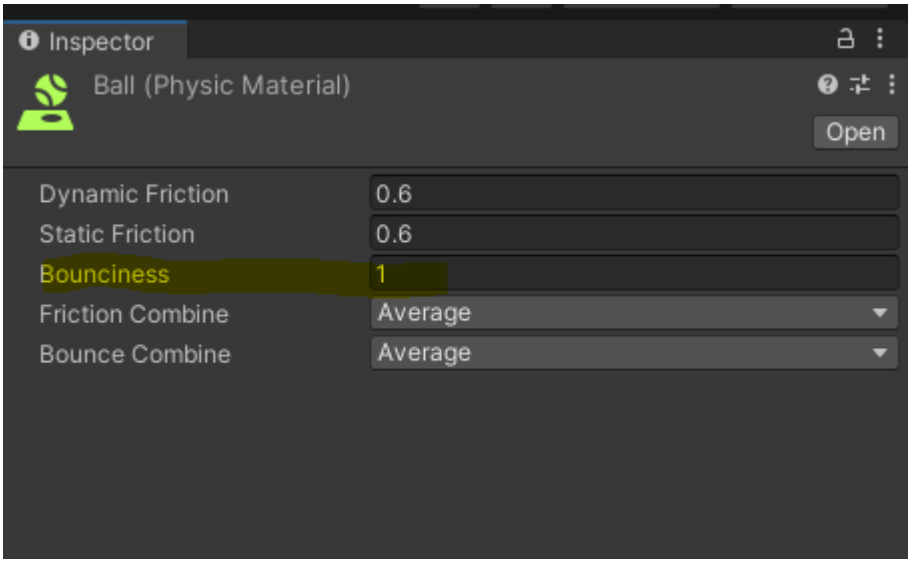
11/10/2022

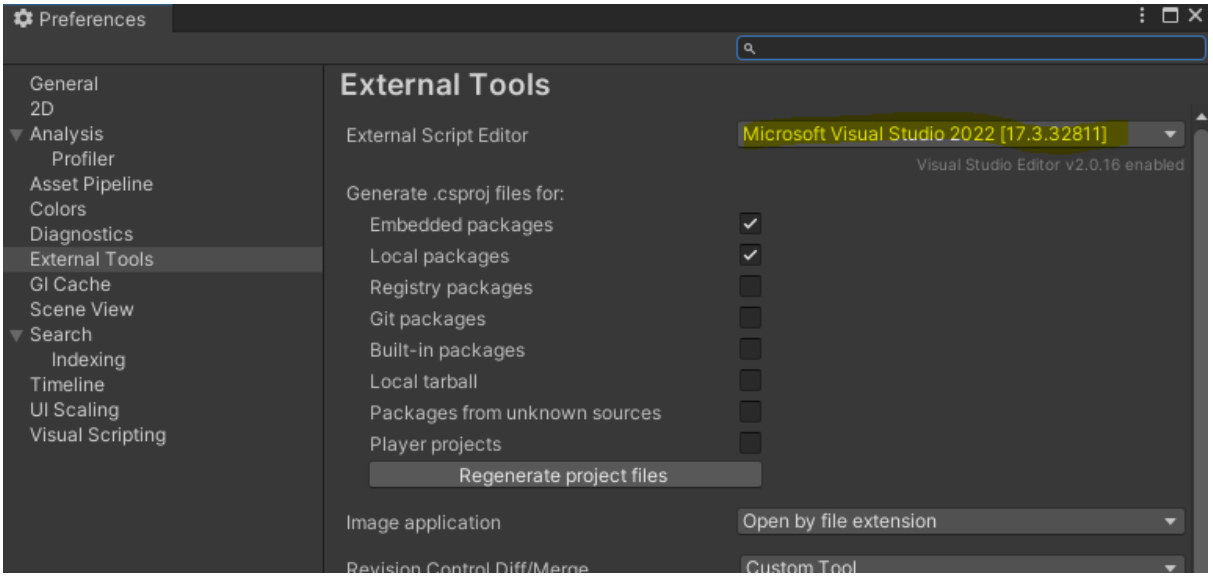
Tutorial was going fine until I had to switch on the Auto Generate option in the Lighting settings. In order to do that I first had to Window > Rendering > Lighting and click the New Lighting Settings under Lighting Settings.

The ball reacts to gravity and falls to the ground, however it doesn't bounce back up. That's because we have to create and add a Physic Material to our ball. A Physic Material allows us to change the values of friction and bouncing to the colliding object it will be added to.



By the default the value would be 0.

My AutoComplete wasn't working, to fix it I had to go to Edit > Preferences > External Tools and change External Script Editor from "open by file extension" to "Microsoft Visual Studio".



18/10/2022
Input Axis Horizontal is not set up, I spelled "Horizontal" without a t.
When I pressed the W key the player would move backwards, I realised that he was actually moving forwards and I just had to fix the position of the shades.
Camera doesn't switch. Ok, it looks like the problem was that I created an empty game object and then assigned a vertical camera to it, which didn't work. I've now created a virtual camera in the hierarchy directly and it works. Another problem is that when the game starts and the player finds himself in the range of cam1 it will switch to cam2. I fixed this by changing the priority on both cams to 1. However the transition between the two cams is too smooth and fixed this by going to the main camera and the default blend in the cinemachine brain to "Cut".

08/11/2022
Everything seems to be working fine, apart from issues, I may have to play around with the variables such as range etc… It seems like the enemy AI is capable of seeing through walls and manages to catch the player even if they are hiding behind a wall.

15/11/2022
To fix the issue above, I created a layer mask for the walls, which I named obstacleMask. I then created a new method in the script called EnviromentView(), this method pretty much allows the enemy to be able to detect the player and the obstacle. The first bit of the method creates a sphere around the enemy which allows it to detect the player and if the player in the radius of the enemy view the player's position will be registered. The else statement checks if the player is behind an obstacle, if that's true the player's position will not be registered and the Enemy AI will not chase them.

```
1 reference
void EnviromentView()
{
    Collider[] playerInRange = Physics.OverlapSphere(transform.position, viewRadius, playerMask);

    for(int i = 0; i < playerInRange.Length; i++)
    {
        Transform player = playerInRange[i].transform;
        Vector3 dirToPlayer = (player.position - transform.position).normalized;
        if(Vector3.Angle(transform.forward, dirToPlayer) < viewAngle / 2)
        {
            float dstToPlayer = Vector3.Distance(transform.position, player.position);
            if(!Physics.Raycast(transform.position, dirToPlayer, dstToPlayer, obstacleMask))
            {
                m_PlayerInRange = true;
                m_IsPatrol = false;
            }

            else
            {
                m_PlayerInRange = false;
            }
        }

        if(Vector3.Distance(transform.position, player.position) > viewRadius)
        {
            m_PlayerInRange = false;
        }

        if (m_PlayerInRange)
        {
            m_PlayerPosition = player.transform.position;
        }
    }

}
```