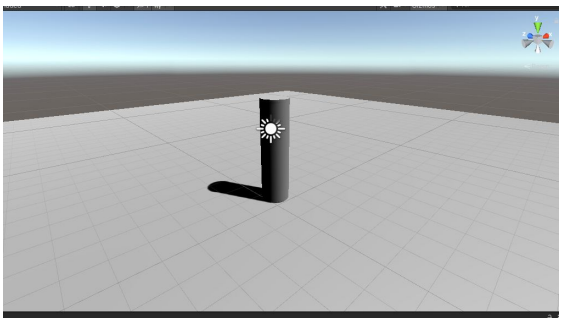
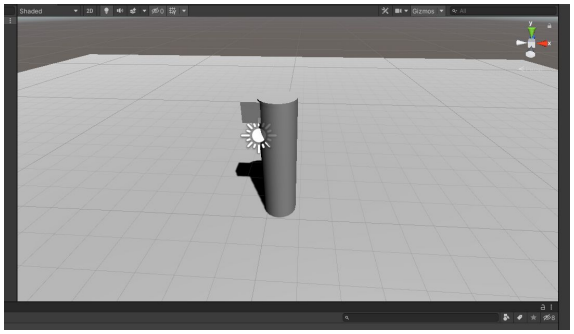
Third Person Movement

Create a new scene in Unity
In the hierarchy create a 3D object plane

In the hierarchy create a 3D object Cylinder to represent the character.
Reset the transform, level the object to the plane and change the scale of the object similar to what the character would be like.



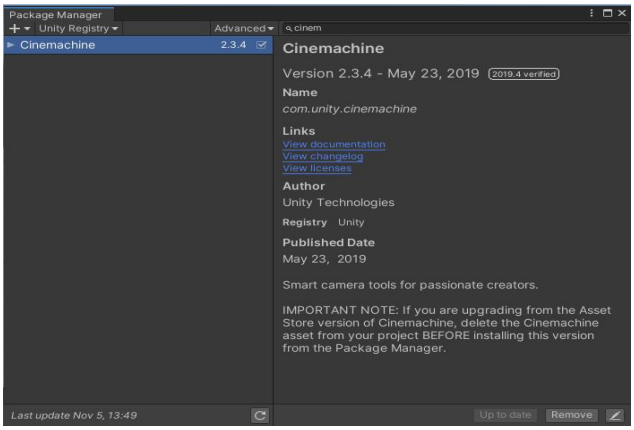
Create another 3D object Cube to indicate the direction of the character.



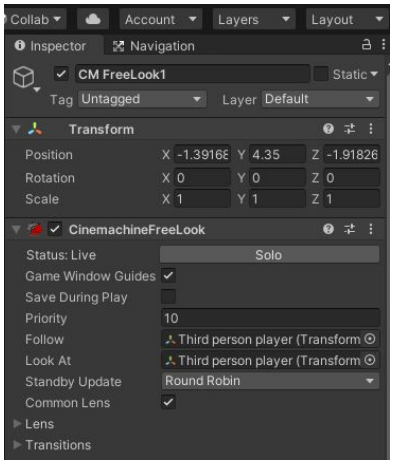
Remove the colliders from both of the objects .

Cinema machine camere controller

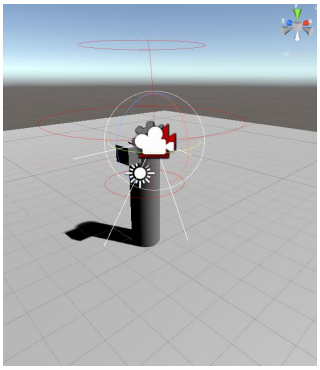
To create the camera controller
In the package manager search for Cinemachine and install it .

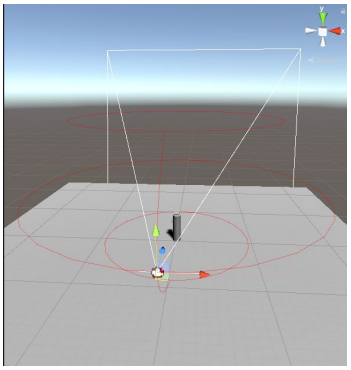


Create a freelook camera, as we want the camera to follow the player we set the camera to follow the player in the inspector panel by dragging the player object over.
We also want the camera to look at the player so we drag the player object over the Look at.



To achieve the camera view as we want we have to change the values of the Orbits of the camera.
Change the binding mode to World Space, we don't want the camera to be influenced by the rotation of the character, we want the character to follow the direction of the camera.





Character movement

Select the player and add component- Character controller
Change the radius to 0.6 and the height to 3.6

Add another component : ThirdPersonMovement script
Remove the start option
Reference the character controller and name it controller;
Create a speed variable and set it to 6f;

To move the character using vast and arrow keys.

In the update section and a horizontal and vertical input to be able to control the character.
Create a vector3 to store the direction, setting the y axis as 0f so the player doesn't move in the y axis .
To make sure we don't that when we move two keys diagonally too fast we have to normalise the vector direction.

To apply the movement we need to define that when the direction and magnitude is higher than 0.1f we get the input to move.

To point the player to the direction it's moving we need to take our direction and decide how much we should rotate the player in the y axis using the Atan2 function.
Atan2 is a mathematical function that returns the angle between the axis with the vector starting at 0 and ending up at x,y.
To set the rotation to the player angle we use the Quaternion.Euler function.
If the player snaps quickly into place we need to add smoothing by creating a variable and use the function SmoothDampAngle, where we include the current angle, the target angle and we need a variable that will hold the smooth velocity so we create a private variable to define it and input the smooth time.

Now when we change the targetAngle to our new angle the player will turn smoothly.

To make the player travel to the direction that the camera is facing we need to reference our camera.
Add the rotation of the camera on the y axis in the calculation of our target angle .

To make sure we move to the direction of the camera we need to create a new vector3 called moveDir, to transform the rotation to movement we multiply it by Vector3.forward.
Change the direction of the controller to moveDir and normalise it.
To apply we need to set the script camera to our Main Camera.

With this script if we move the mousse a lot we are able to look everywhere it's possible to lose the player behind another object or move through an object, to avoid this to happen Cinemachine as a built-in feature, for this we go to our Cinemachine camera and add a CinemachineCollider and select the layer where the objects are to collide against., change the strategy to Pull Camera Forward (this will pull the camera towards the player when colliding with other objects)

Full Script

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Collections.Specialized;
using System.Diagnostics;
using System.Security.Cryptography;
using System.Threading;
using UnityEngine;


public class ThirdPersonMovement : MonoBehaviour
{

    public CharacterController controller;
    public Transform cam;

    public float speed = 6f;
    public float jumpHeight = 5f;

    public float turnSmoothTime = 0.1f;
    float turnSmoothVelocity;


    // Update is called once per frame
    void Update()
```

```
    {
        Jump();
        float horizontal = Input.GetAxisRaw("Horizontal");
        float vertical = Input.GetAxisRaw("Vertical");
        Vector3 direction = new Vector3(horizontal, 0f, vertical).normalized;

        if (direction.magnitude >= 0.1f)
        {
            float targetAngle = Mathf.Atan2(direction.x, direction.z) * Mathf.Rad2Deg +
cam.eulerAngles.y;
            float angle = Mathf.SmoothDampAngle(transform.eulerAngles.y, targetAngle,
ref turnSmoothVelocity, turnSmoothTime);
            transform.rotation = Quaternion.Euler(0f, angle, 0f);

            Vector3 moveDir = Quaternion.Euler(0f, targetAngle, 0f) * Vector3.forward;
            controller.Move(moveDir.normalized * speed * Time.deltaTime);

        }


    }
    void Jump()
    {

        if (Input.GetButton("Jump"))
        {
            UnityEngine.Debug.Log("I'm jumping");
             controller.GetComponent<Rigidbody>().AddForce(0, jumpHeight, 0,
ForceMode.Impulse);

        }
    }

}
```