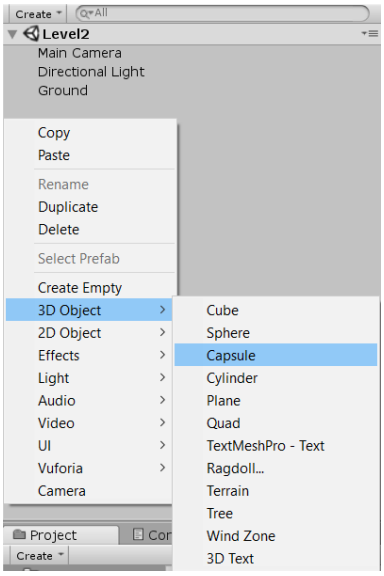
First Person Controller

Here is a tutorial on how to program a first-person controller from scratch, please note I am assuming you have read my previous tutorials and know "[ASB]" means 'As Shown Below'. I hope you know what **Hierarchy**, **Assets** Folder and **Inspector** are in Unity too.

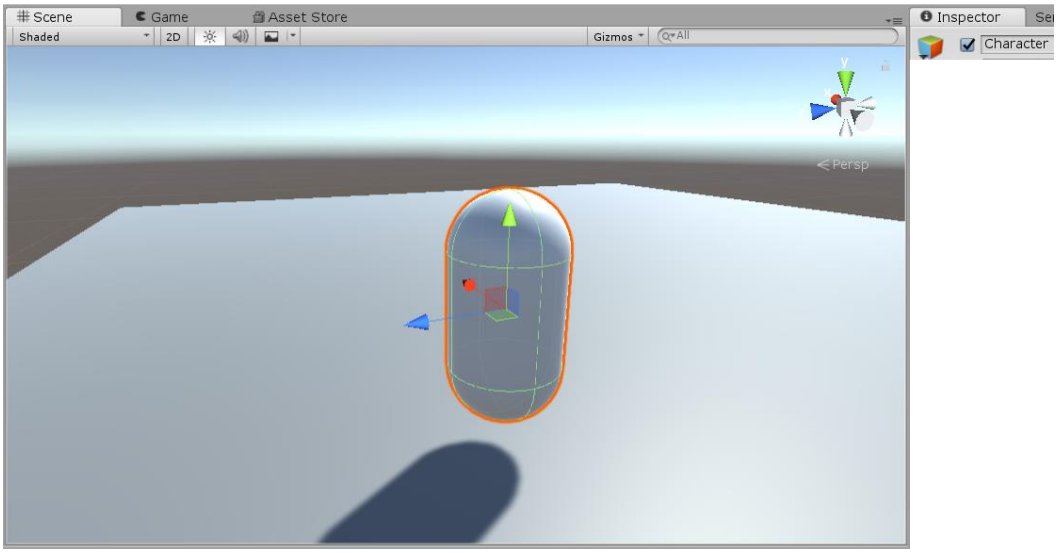
To make your life easier: Save your work after each step.

Step 1: Open **Unity** and load the scene that you want to edit, (I am assuming you already have a base "Ground" model with **Collider**)

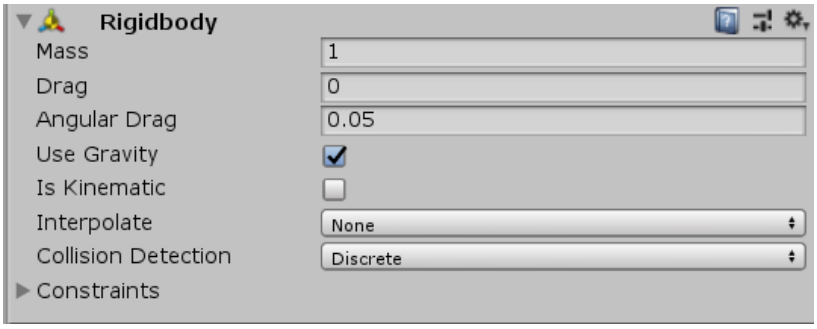
Step 2: Right Click on your **Hierarchy** > **3D Object** > **Capsule** [ASB] This will be the player



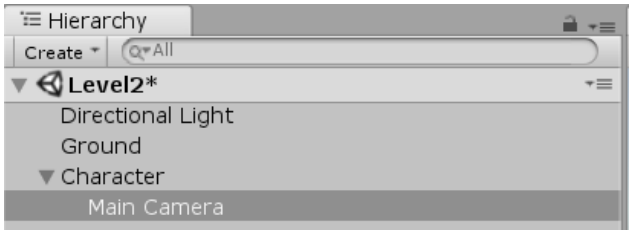
To make your life easier: Make sure it is above your ground and if you like, rename it to **Character** [ASB]



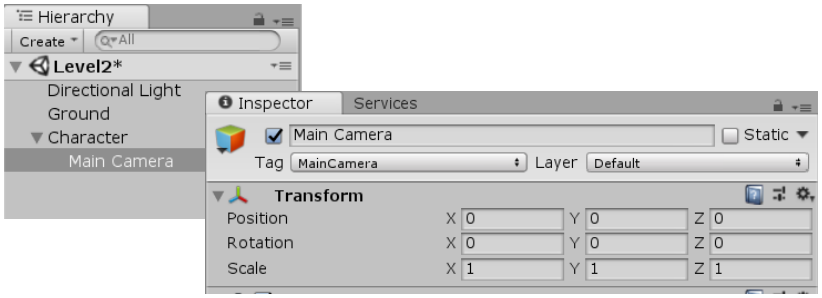
Step 3: Click on Character, under its Inspector, Add **Component** > **Rigidbody** [ASB]



Step 4: Grab **Main Camera** in Hierarchy and Drop it onto **Character**, we want them to be connected [ASB]

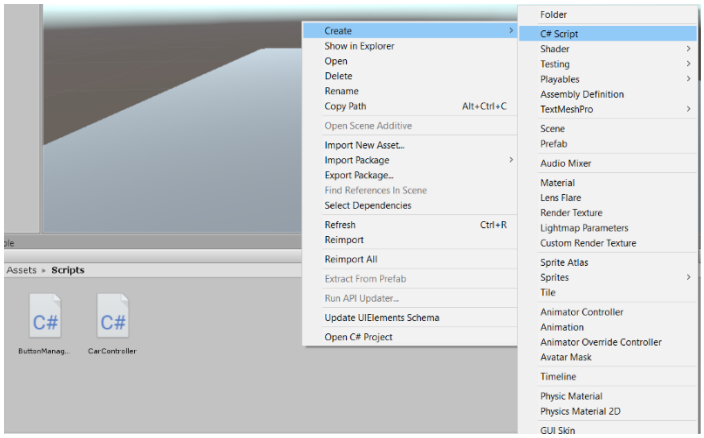


Step 5: Click on the **Main Camera**, change its **Position** to *X = 0, Y = 0, Z = 0* [ASB]



To make your life easier: Make a **Folder** in Your **Asset File** and call it **Scripts**.

Steps 6: Right Click in your **Scripts File** > **Create** > **C# Script** [ASB] and call it ***CharacterController***



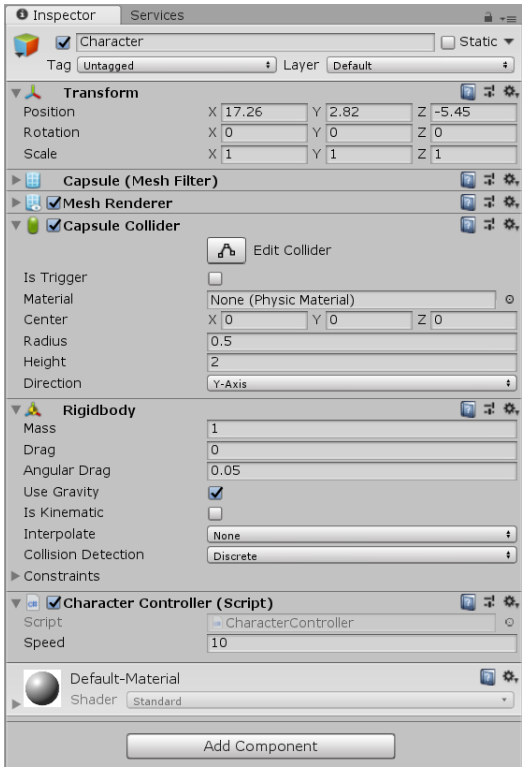
Step 7: Open the new created script, and write the following script: (pay attention to Comments, they explain the code for your better understandings:

CharacterController.cs*

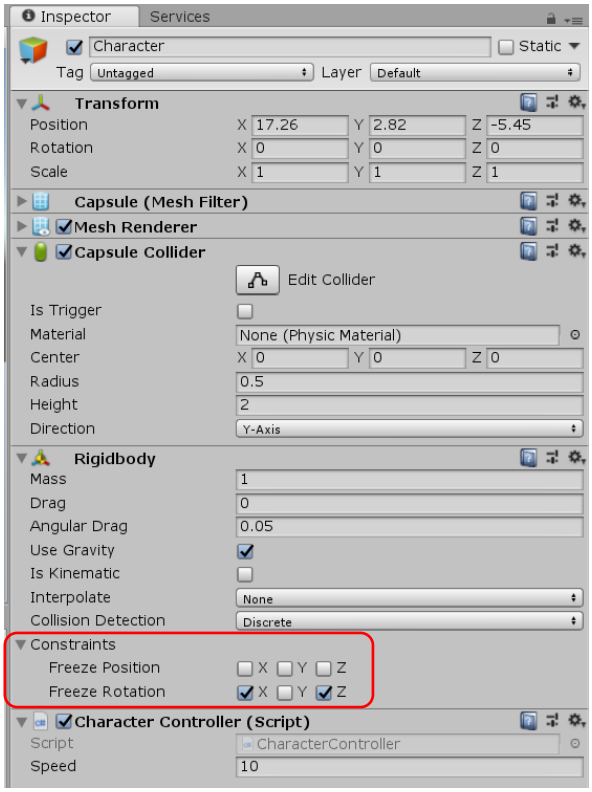
Toturials — CharacterController

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class CharacterController : MonoBehaviour {

    public float speed = 10f; //speed of respond

    // Use this for initialization
    void Start()
    {
        Cursor.lockState = CursorLockMode.Locked; // so you dont see the cursor and keeps it in the game box
    }

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

        float translation = Input.GetAxis("Vertical") * speed;
        float straffe = Input.GetAxis("Horizontal") * speed;
        float jump = Input.GetAxis("Jump");
        translation *= Time.deltaTime;//keeps movement smooth and in time with update
        straffe *= Time.deltaTime;

        transform.Translate(straffe, 0, translation); //translation is moving forward and backword

        if (Input.GetKeyDown("escape"))
            Cursor.lockState = CursorLockMode.None; //brings the cursor back when esc is pressed
    }
}
```

Step 8: Save and go back to **Unity**, click on the **Character** and Drag and Drop the ***CharacterController*** Script on to Charater **Inspector**. [ASB]



Step 9: Select **Character**, look at the **Inspector**, under the **Rigitbody** > **Constraints** > Make sure Freeze Rotation X and Z are ticked [ASB]

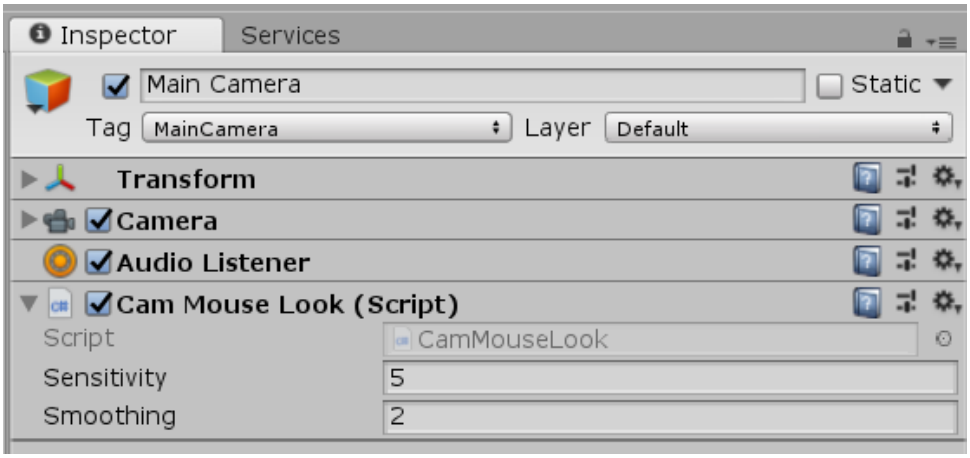


Step 10: Go to your **Script File** and Right Click > **Create** > **C# Script** and name it ***CamMouseLook***

Step 11: Open the **Script** to edit and add the following code:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class CamMouseLook : MonoBehaviour {

    Vector2 mouselook;// how much movement the camera have made (total)
    Vector2 smoothV;// smooth down the movement of the mouse
    public float sensitivity = 5.0f;// mouse sensetivity
    public float smoothing = 2.0f;// how much smoothing do you want?

    GameObject character;

    // Use this for initialization
    void Start()
    {
        character = this.transform.parent.gameObject;// the parent that the object is connected to is the character
    }

    // Update is called once per frame
    void Update()
    {
        var md = new Vector2(Input.GetAxisRaw("Mouse X"), Input.GetAxisRaw("Mouse Y"));

        smoothV.x = Mathf.Lerp(smoothV.x, md.x, 1f / smoothing);
        smoothV.y = Mathf.Lerp(smoothV.y, md.y, 1f / smoothing);
        mouselook += smoothV;
        mouselook.y = Mathf.Clamp(mouselook.y, -90f, 90f); // clampin the looking up and down

        transform.localRotation = Quaternion.AngleAxis(-mouselook.y, Vector3.right);
        character.transform.localRotation = Quaternion.AngleAxis(mouselook.x, character.transform.up);
    }
}
```

Step 12: Save and go back to **Unity**, Drag the **Script** from the *Scripts Folder* and Drop it under the **Main Camera** [ASB]



Step 13: Make sure you have **saved** your work and are ready to **move** around.