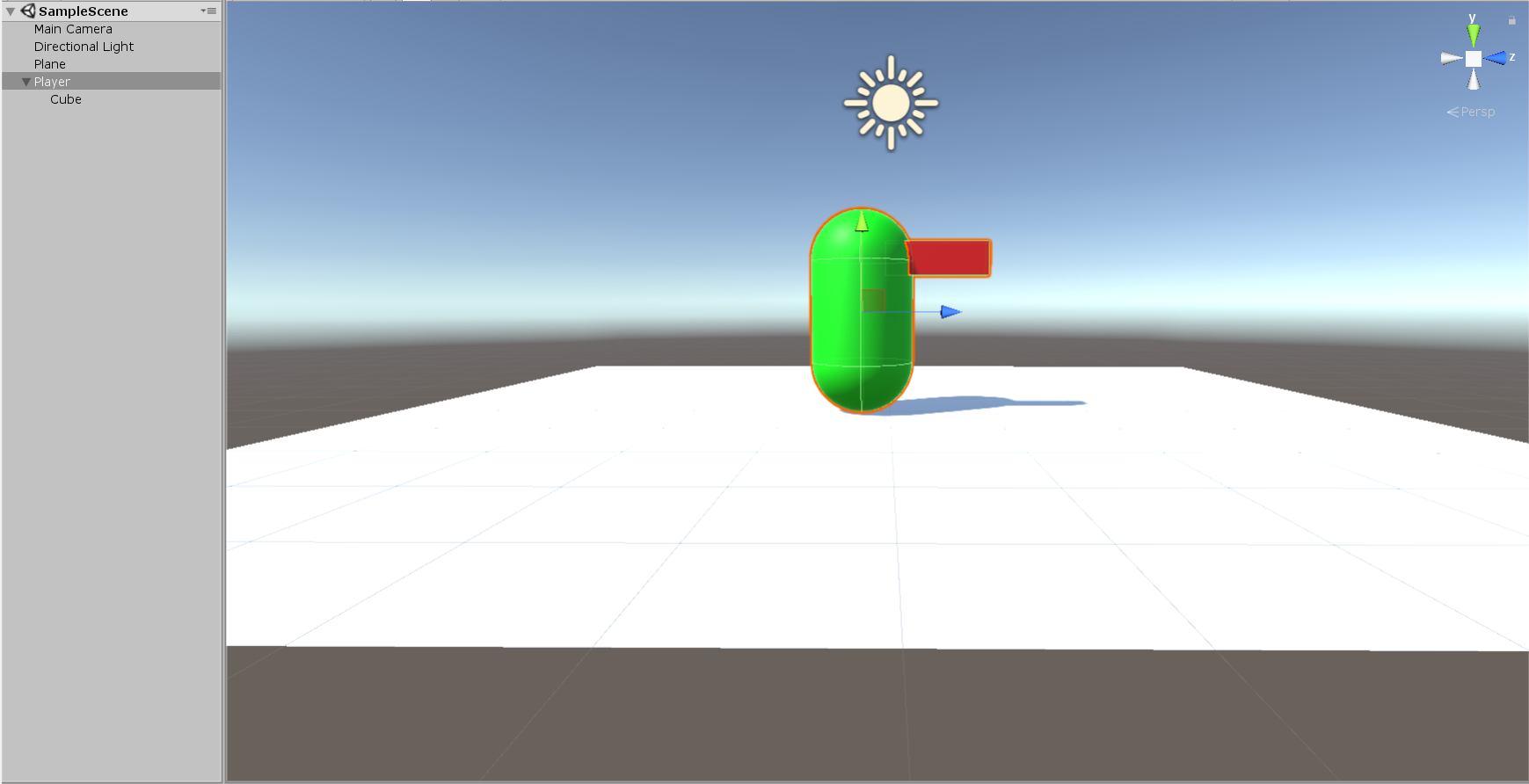
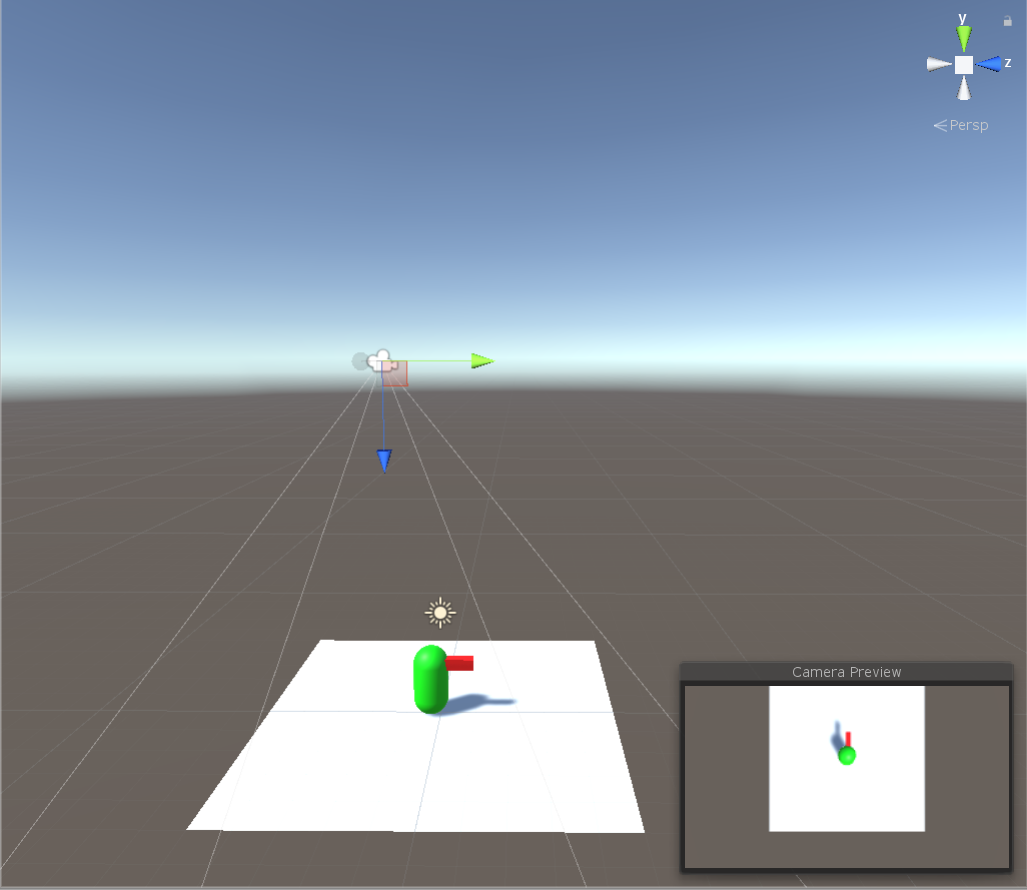
**Behaviour 3 Tutorial**

**This Tutorial will teach you how to move a character and aim with your mouse in a top down view.**

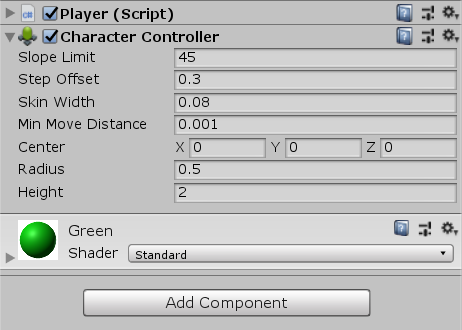
**Step 1: Create a plane, capsule (rename it player) and a cube. Scale the cube so it is a short rectangle. You can assign colours to each of the object if you like. In the inspector, click and drag your cube into your player so the cube becomes a child of the player. Then position each object as seen in the image below.**



**Step 2: Position your camera so it looks down on the player.**

****

**Step 3: Add a Character Controller component to your player and add a script to your player called Player.**

****

**Step 4: Open your player script.**

**Declare the 4 main variables we need:**

****

**Step 5: Under Void Start.**

**Enter: player = GetComponent <CharacterController>(); ``This will get the character component from our player.**

**mainCamera = FindObjectOfType<Camera>(); ``This will find thecamera in our scene.**

**Step 6: Under Void Update, we are going to add the script for the movement of our character and the code for aiming with our mouse.**

  Vector3 Move = Vector3.zero; ``we are declaring a vector 3 variable called move and setting it to 0.  
        Move.x = Input.GetAxis ("Horizontal") \* Speed; ``This gets the horizontal control input and converts them to our vector 3 move x axis and times it by speed.  
        Move.z = Input.GetAxis ("Vertical") \* Speed; ``This gets the vertical control input and converts them to our vector 3 move z axis and times it by speed.  
        player.Move(Move \* Time.deltaTime); ``This will apply our input to our player so it will move.  
  
        Ray cameraRay = mainCamera.ScreenPointToRay (Input.mousePosition); ``We are declaring a variable called ray, the value will be the position of our mouse relative to the camera screen space.  
  
        Plane ground = new Plane (Vector3.up, Vector3.zero); ``Here we are declaring a variable called plane and call it ground. This will be used to detect the direction our ray hits.  
  
        if (ground.Raycast(cameraRay, out rayLength)) { ``This detects if the ray we cast from our camera will hit the ground and returns the ray length.  
  
            Vector3 lookTarget = cameraRay.GetPoint (rayLength); ``We declare a new vector 3 variable, it will be used to store the tip of the point our ray hits.  
  
            transform.LookAt (new Vector3(lookTarget.x, transform.position.y, lookTarget.z)); ``This will tell the player to look at the vector 3 we just created which contains the point in which our ray hits.  
        }

**Step 7: Set your speed in the inspector and press play. You will be able to move with the WASD button and aim with your mouse.**