

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class FirstPersonController : MonoBehaviour
6 {
7     // Variables for movement and interaction
8     public float movementSpeed = 5.0f;
9     public float mouseSensitivity = 5.0f;
10    public KeyCode interactKey = KeyCode.E;
11
12    /*
13     * The reason i had a groundcheck was issues with the player falling ↗
14     * over, this however
15     * is more likely due to approaching a slope and the game not having ↗
16     * a navmesh agent, this needs more
17     * monitoring and possible fix. (Meanwhile; AVOID SLOPES!)
18     */
19
20    // Ground check
21    //public float gravity = -9.81f;
22    //public Transform groundCheck;
23    //public float groundDistance = 0.4f;
24    //public LayerMask groundMask;
25    //bool isGrounded;
26
27    //Vector3 velocity; // Not sure about this line but i am using it ↗
28    //for ground checking
29
30    private Camera playerCamera;
31    private Rigidbody rb;
32
33    void Start()
34    {
35        playerCamera = GetComponentInChildren<Camera>();
36        rb = GetComponent<Rigidbody>();
37        if (rb == null)
38        {
39            rb = gameObject.AddComponent<Rigidbody>();
40        }
41
42        // Lock and hide the cursor
43        Cursor.lockState = CursorLockMode.Locked;
44        Cursor.visible = false;
45    }
46
47    void Update()
48    {
49        // Ground function
50        //isGrounded = Physics.CheckSphere(groundCheck.position, ↗
51        //    groundDistance, groundMask);
52        //if (isGrounded && velocity.y < 0)
53        //{
```

```
50     //    velocity.y = -2f;
51     //}
52
53     //velocity.y += gravity * Time.deltaTime;
54
55     // Get input for movement and interaction
56     float horizontal = Input.GetAxis("Horizontal");
57     float vertical = Input.GetAxis("Vertical");
58     float mouseX = Input.GetAxis("Mouse X");
59     float mouseY = Input.GetAxis("Mouse Y");
60     bool interact = Input.GetKeyDown(interactKey);
61
62     // Rotate the camera with mouse input
63     transform.Rotate(Vector3.up * mouseX * mouseSensitivity);
64     playerCamera.transform.Rotate(Vector3.right * -mouseY * mouseSensitivity);
65
66     // Move the player
67     Vector3 movement = transform.forward * vertical +
68     transform.right * horizontal;
69     movement = movement.normalized * movementSpeed * Time.deltaTime;
70     rb.MovePosition(rb.position + movement);
71
72     // Raycast to detect and interact with objects
73     if (interact)
74     {
75         RaycastHit hit;
76         if (Physics.Raycast(playerCamera.transform.position,
77         playerCamera.transform.forward, out hit, 2.0f))
78         {
79             // Check if the object can be interacted
80             InteractableObject obj =
81             hit.collider.GetComponent<InteractableObject>();
82             if (obj != null)
83             {
84                 obj.Interact();
85             }
86         }
87     }
```