PLAYER CONTROLLER

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

public class PlayerController : MonoBehaviour {

    public Rigidbody player;
    public float speed;
    public Text countText;
    public Text winText;

    private int count;

    // Use this for initialization
    void Start () {
        player = GetComponent<Rigidbody>();
        count = 0;
        countText.text = "Score:- 0";
    }

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

    }

    // FixedUpdate is called before any Physics Calculations
    void FixedUpdate() {
        float moveHoriz = Input.GetAxis("Horizontal");
        float moveVer = Input.GetAxis("Vertical");
        Vector3 move = new Vector3(moveHoriz,0f,moveVer) * speed;
        player.AddForce(move);
    }

    void OnTriggerEnter(Collider other) {
        if(other.gameObject.CompareTag("pickup")) {
            other.gameObject.SetActive(false);
            count+=10;
            countText.text = "Score:- "+ count.ToString();
            if(count>=80) {
                winText.gameObject.SetActive(true);
            }
        }
    }

    public void Reset() {
        Application.LoadLevel(Application.loadedLevel);
    }
}
```

CAMERA CONTROLLER

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

public class CameraController : MonoBehaviour {

    public GameObject player;

    private Vector3 offset;

    // Use this for initialization
    void Start () {
        offset = transform.position - player.transform.position;
    }

    // Update is called once per frame
    void LateUpdate () {
        transform.position = player.transform.position + offset;
    }
}
```

ROTATOR

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

public class Rotator : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {
        transform.Rotate (new Vector3 (15, 30, 45) * Time.deltaTime);
    }

}
```