

```

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

public class PlayerMovement : MonoBehaviour
{
    // ShortCuts
    Rigidbody rb;

    //Side Movement
    public KeyCode keyPositive;
    public KeyCode keyNegetive;
    public Vector3 Direction;
    public float side;

    //Height Movement
    public float jumpHeight;
    bool InAir = false;
    [SerializeField]
    string groundTag;

    //Speed Movement
    public float startSpeed;
    public float accelSpeed;
    public float maxSpeed;

    //Death
    [SerializeField]
    string collisionTag;
    public GameObject deathEfx;
    Vector3 expoPlace;

    void Start()
    {
        //Setting Shortcut
        rb = GetComponent<Rigidbody>();

        //Setting Speed Movement
        rb.velocity = new Vector3(0f, 0f, startSpeed);
        rb.maxAngularVelocity = Mathf.Infinity;
    }
}

```

```

void Update()
{
    Vector3 movement = new Vector3(0f, 0f, 0f);
    // Setting Side Movement
    if (Input.GetKey(keyPositive))
        //rb.velocity += Direction;
        rb.velocity += (new Vector3(side, 0f, 0f)+movement)*Time.deltaTime;
    if (Input.GetKey(keyNegative))
        //rb.velocity -= Direction;
        rb.velocity += (new Vector3(-side, 0f, 0f)+movement)*Time.deltaTime;

    // Setting Height Movement
    if (Input.GetButtonDown("Jump") && InAir == false)
    {
        rb.AddForce((new Vector3(0.00001f, jumpHeight, 0.00001f)+movement),
ForceMode.Impulse);
        InAir = true;
    }
}

```

```

void FixedUpdate()
{
    //Setting Speed Movement

    rb.velocity += new Vector3(0f, 0f, accelSpeed)*Time.deltaTime;

    if (rb.velocity.magnitude > maxSpeed)
    {
        rb.velocity = Vector3.ClampMagnitude(rb.velocity, maxSpeed);
    }
}

```

```

void OnCollisionEnter(Collision other)
{
    //Setting Height Movement
    if (other.gameObject.tag == groundTag && InAir == true)
    {
        InAir = false;
    }
}

```

```
//Setting Death
if (other.collider.tag == collisionTag)
{
    expoPlace = transform.position;
    Instantiate(deathEfx, expoPlace, Quaternion.identity);
    Destroy(this.gameObject);
}
}
```