



Your version of Visual Studio is no longer supported or serviced. Update to the latest version of 17.14 to remain supported and receive the latest security and quality fixes or install Visual Studio 2026. Update 17.14 Install Visual Studio 2026

```

Archivos varios
PlayerMove.cs PlayerMoveJoystick.cs
Archivos varios
using System.Collections;
using System.Linq;
using UnityEngine;

public class PlayerMoveJoystick : MonoBehaviour
{
    [Header("Componentes")]
    public Joystick joystick; // Asocia el variante Joystick aquí
    public Spriterenderer spiroRender; // Asocia el Spriterenderer aquí
    public Animator animator; // Asocia el Animator aquí
    private Rigidbody2D rb2D;

    [Header("Configuración de Movimiento")]
    public float velocidad = 5f;
    public float jumpSpeed = 8f;

    [Header("Atributos Privados")]
    public bool betterJump = true;
    public float velocidadMultiplicador = 1.5f;
    public float tanqueMultiplicador = 2f;

    [Header("Transiciones")]
    public float vida = 3;

    private float horizontalInput;
    private bool canDoubleJump;

    void Start()
    {
        rb2D = GetComponent();
    }

    // VALIDACIÓN POR SI ALGUNOS ARRASTRADOS EN EL INSPECTOR
    if (animator == null) animator = GetComponent();
    if (spiroRender == null) spiroRender = GetComponent();
}

void Update()
{
    // INPUT DEL JOYSTICK
    if (joystick != null)
    {
        horizontalInput = joystick.Horizontal;
    }

    // SI SE PULSA EL BOTÓN
    if (joystick.button == 0.1f)
        spiroRender.flipX = false;
    else if (joystick.button == -0.1f)
        spiroRender.flipX = true;

    // ACTUALIZAR ANIMACIONES
    UpdateAnimations();
}

void FixedUpdate()
{
    // MOVIMIENTO FÍSICO
    rb2D.velocity = new Vector2(horizontalInput * velocidad, rb2D.velocity.y);
}

```

Linea: 140 Carácter: 2 SPC CRLF

The screenshot shows the Visual Studio 2026 IDE interface. The menu bar includes Archivo, Editar, Ver, Git, Proyecto, Depurar, Prueba, Analizar, Herramientas, Extensiones, Ventana, Ayuda, Buscar, and Solución. A status bar at the bottom indicates "64 %", "No se encontraron problemas.", "Linea: 140", "Carácter: 2", "SPC", and "CR/LF". The main window displays the code for PlayerMove.cs, which contains logic for player movement and physics, including methods like ExecuteJump, ExecuteSecondJump, and UpdateAnimation. The code uses Unity's Physics2D API and Animator components. The right side of the screen shows the Solution Explorer, Project Explorer, and Task List panes.

```
private void ExecuteJump()
{
    rb2D.velocity = new Vector2(rb2D.velocity.x, jumpSpeed);
    canDoubleJump = true;
}

private void ExecuteSecondJump()
{
    rb2D.velocity = new Vector2(rb2D.velocity.x, jumpSpeed);
    canDoubleJump = false;
    animator.SetBool("DoubleJump", true);
}

private void OnCollisionEnter2D(Collision2D collision)
{
    if (collision.gameObject.CompareTag("Obstacle"))
    {
        vida -= 1;
        Destroy(collision.gameObject);
        Debug.Log("Vida restante: " + vida);
    }
}

void UpdateAnimation()
{
    bool isGrounded = CheckGround();
    float verticalVelocity = rb2D.velocity.y;

    // Correr: Se activa si el jugador se mueve lo suficiente y toca suelo
    animator.SetBool("Run",Mathf.Abs(horizontalInput) > 0.1f && isGrounded);
    // Saltar
    if (!isGrounded && verticalVelocity < -0.1f)
    {
        animator.SetBool("Falling", true);
        animator.SetBool("DoubleJump", false);
        animator.SetBool("Jump", false);
    }
    else animator.SetBool("Falling", false);

    // Saltar
    if (!isGrounded && verticalVelocity > 0.1f && animator.GetBool("DoubleJump"))
    {
        animator.SetBool("Jump", true);
    }
    else if (isGrounded || animator.GetBool("Falling"))
    {
        animator.SetBool("Jump", false);
    }
}

if (isGrounded) animator.SetBool("DoubleJump", false);
```