

The screenshot shows the Unity Editor's code editor with the file `MovementController.cs` open. The code defines a `MovementController` class that inherits from `MonoBehaviour` . It includes methods for `Start()`, `Update()`, and `FixedUpdate()`. The `Update()` method uses `Input.GetAxis` to get horizontal and vertical movement inputs, then scales them by `movementSpeed` and applies them to the `Rigidbody2D` component. The `FixedUpdate()` method is used for physics calculations.

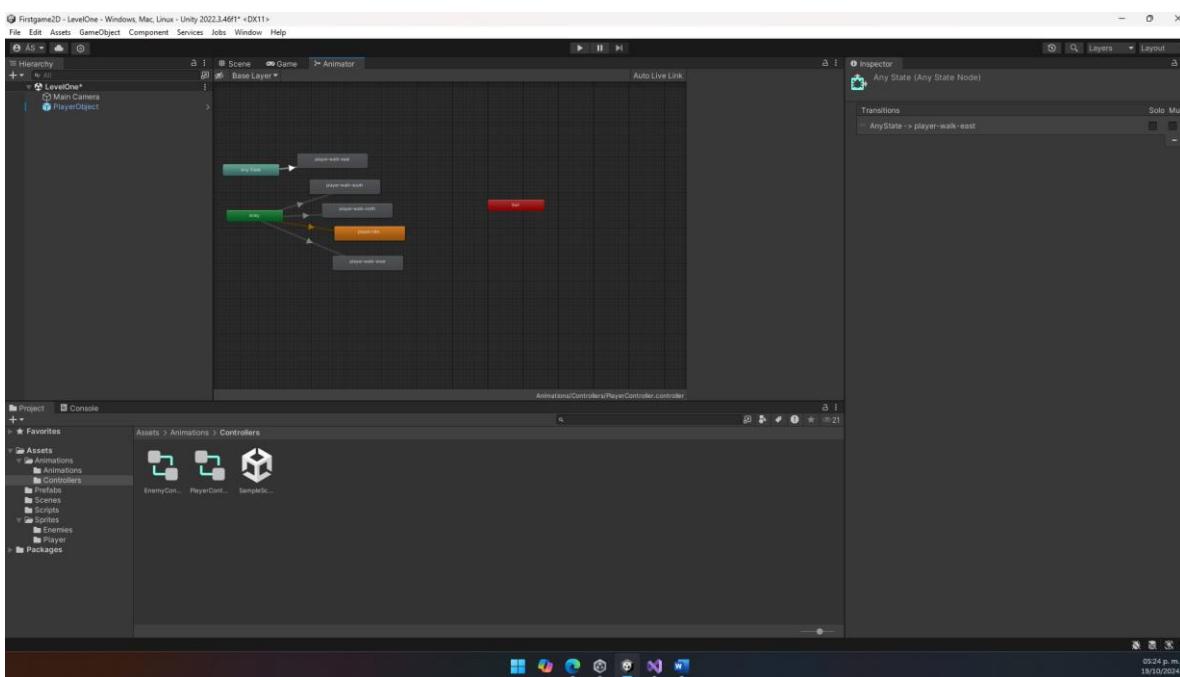
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

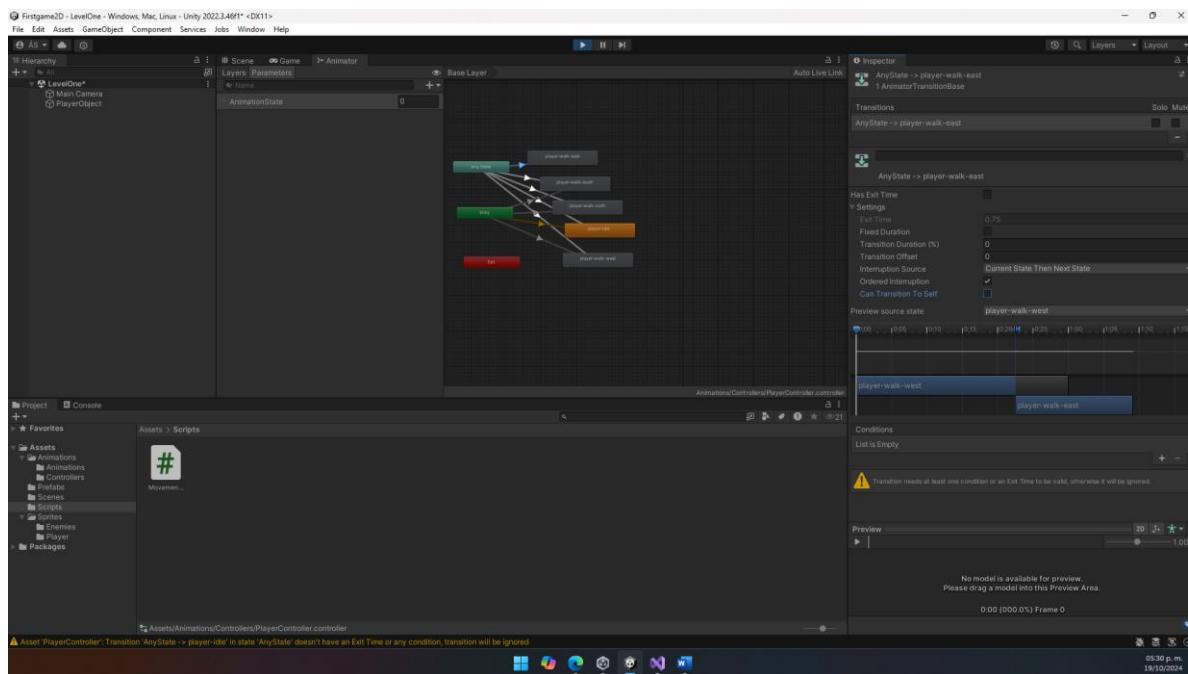
public class MovementController : MonoBehaviour
{
    public float movementSpeed = 3.0f;
    Vector2 movement = new Vector2(0, 0);

    // Start is called before the first frame update
    void Start()
    {
        rb2D = GetComponent();
    }

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

    void FixedUpdate()
    {
        movement.x = Input.GetAxis("Horizontal");
        movement.y = Input.GetAxis("Vertical");
        // Conservar la velocidad
        movement.Normalize();
        rb2D.velocity = movement * movementSpeed;
    }
}
```





The screenshot shows the Unity Editor's code editor window with the file `MovementController.cs` open. The code defines a `MovementController` class that extends `MonoBehaviour`. It includes fields for movement speed, movement vector, and a rigidbody. An animator component is referenced to change animation states based on movement direction. The `Start()` method initializes the rigidbody and animator. The `Update()` method calls `UpdateState()`. The `UpdateState()` method sets the `animationState` integer to different values depending on the movement direction (`walkEast`, `walkSouth`, `walkWest`, `walkNorth`, or `idleSouth`). The Unity interface also shows the project browser with the solution and assembly, and the top bar with tabs like Archivo, Editor, Ver, Git, Proyectos, Compilar, Depurar, Analizar, Herramientas, Extensiones, Vegeta, Ayuda, and Buscar.

```
using System;
using UnityEngine;
using UnityEngine.Animations.Rigidbody;

public class MovementController : MonoBehaviour
{
    public float movementSpeed = 1.6f;
    Vector2 movement = new Vector2(0);
    Rigidbody2D rb2D;

    Animator animator; //Referencia a componente animator
    string animationState = "AnimationState"; //Variable para almacenar el estado de la animación

    enum CharStates
    {
        walkEast = 1,
        walkSouth = 2,
        walkWest = 3,
        walkNorth = 4,
        idleSouth = 5
    }

    void Start()
    {
        //Establecemos el componente Rigidbody2D enlazado
        rb2D = GetComponent<
```

The screenshot shows the Visual Studio IDE interface with the code editor open to the file `MovementController.cs`. The code implements movement logic based on input axes and animates the character accordingly. It includes methods for `update()`, `FixedUpdate()`, and `MoveCharacter()`.

```
using System.Collections; using System.Collections.Generic; using UnityEngine; public class MovementController : MonoBehaviour { void update() { this.UpdateState(); // Invocamos el método UpdateState } private void UpdateState() { if(movement.x > 0) //Este es el estado de caminar hacia el este animator.SetInteger(animationState, (int)CharStates.walkEast); else if (movement.x < 0) //Este es el estado de caminar hacia el oeste animator.SetInteger(animationState, (int)CharStates.walkWest); else if (movement.y > 0) //Este es el estado de caminar hacia el norte animator.SetInteger(animationState, (int)CharStates.walkNorth); else if (movement.y < 0) //Este es el estado de caminar hacia el sur animator.SetInteger(animationState, (int)CharStates.walkSouth); else //Este es el estado de estar quieto animator.SetInteger(animationState, (int)CharStates.idleSouth); } void FixedUpdate() { MoveCharacter(); //Método definido para ingresar la dirección del personaje } private void MoveCharacter() { movement.x = Input.GetAxis("Horizontal"); movement.y = Input.GetAxis("Vertical"); movement.Normalize(); rb2D.velocity = movement * movementSpeed; } }
```

