

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

public class MovementController : MonoBehaviour
{
    // Velocidad de los personajes
    public float movementSpeed = 3.0f;

    // Ubicación player o enemy
    Vector2 movement = new Vector2(0);

    // Referencia a Rigidbody2D
    Rigidbody2D rb2D;

    // Start is called before the first frame update
    void Start()
    {
    }

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

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

public class MovementController : MonoBehaviour
{
    // Velocidad de los personajes
    public float movementSpeed = 3.0f;

    // Ubicación player o enemy
    Vector2 movement = new Vector2(0);

    // Referencia a Rigidbody2D
    Rigidbody2D rb2D;

    // Start is called before the first frame update
    void Start()
    {
        rb2D = GetComponent<Rigidbody2D>();
    }

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

The screenshot shows the Visual Studio code editor with the file `MovementController.cs` open. The code implements a `MovementController` class that extends `MonoBehaviour`. It includes fields for movement speed and a `Rigidbody2D` component. The `Start` method initializes the `Rigidbody2D`. The `FixedUpdate` method updates the movement based on user input (captured via `GetAxisRaw`) and applies it to the `Rigidbody2D`.

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

public class MovementController : MonoBehaviour
{
    // Velocidad de los personajes
    public float movementSpeed = 3.0f;

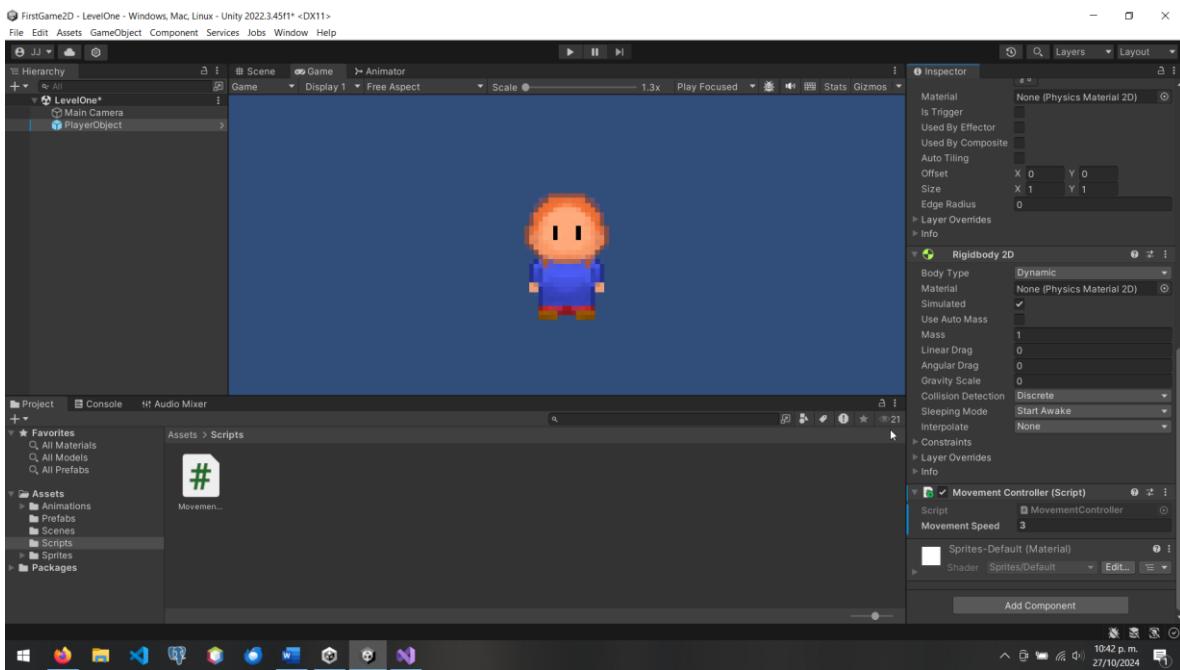
    // Ubicación player o enemy
    Vector2 movement = new Vector2(0, 0);

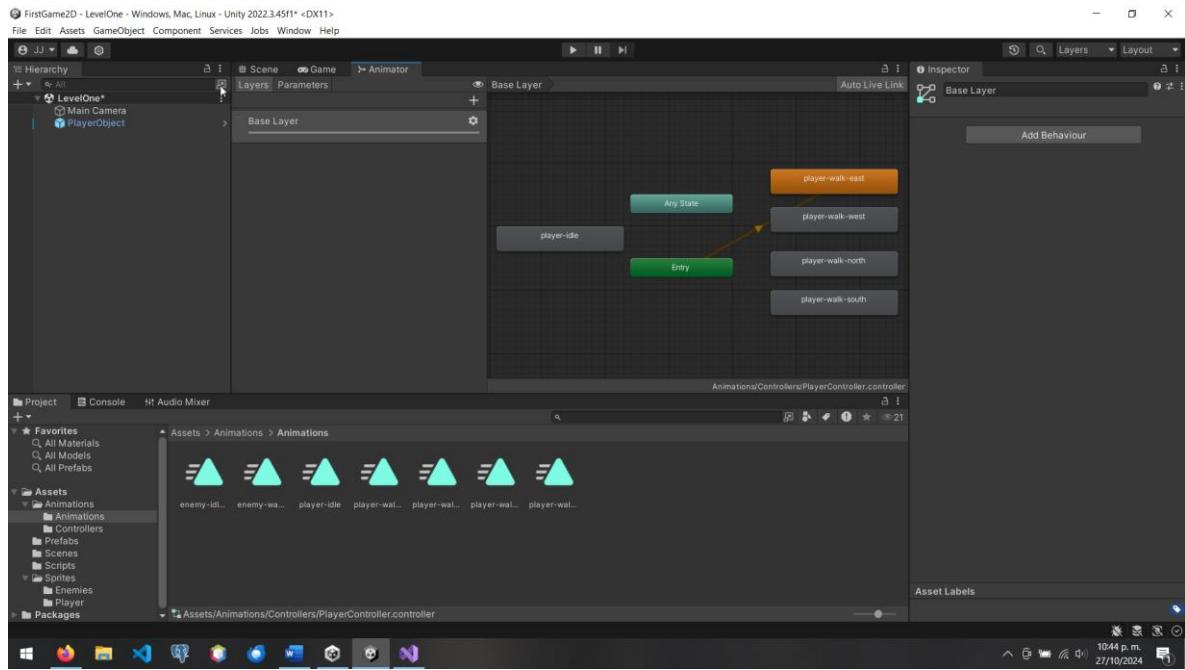
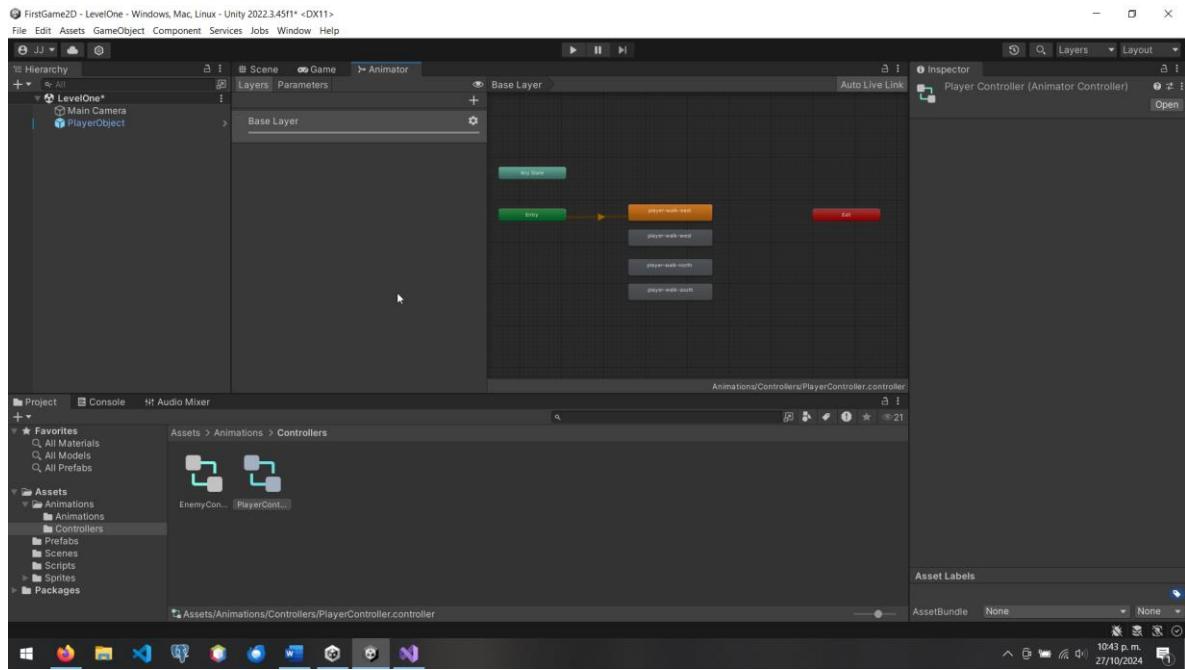
    // Referencia a Rigidbody2D
    Rigidbody2D rb2D;

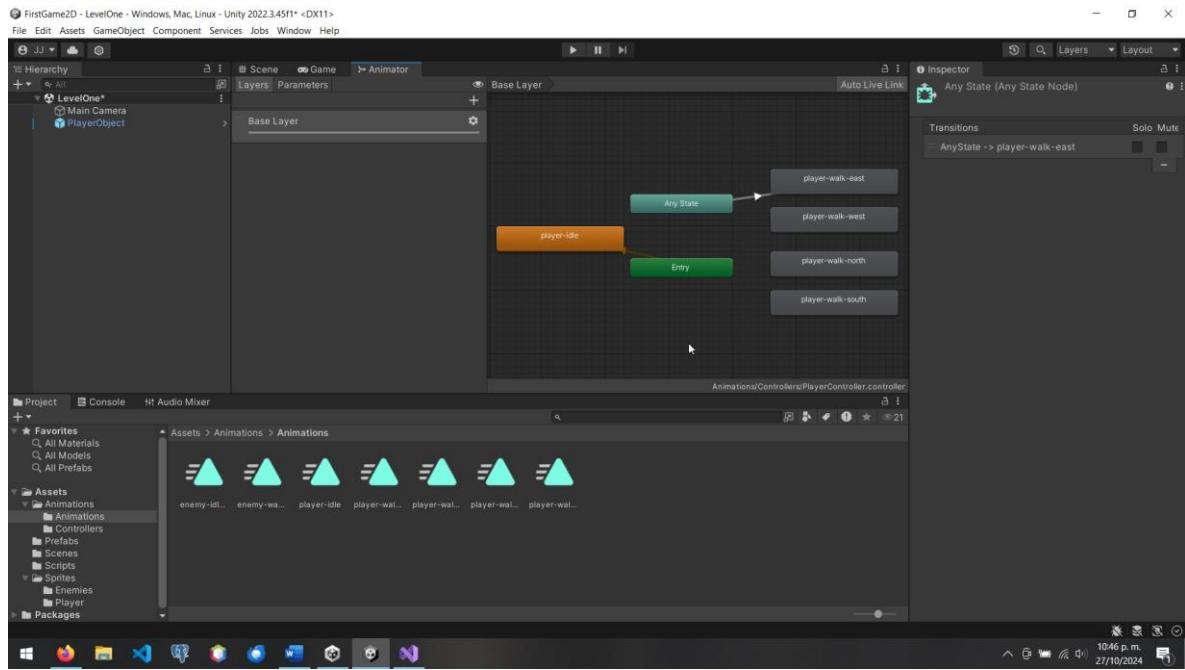
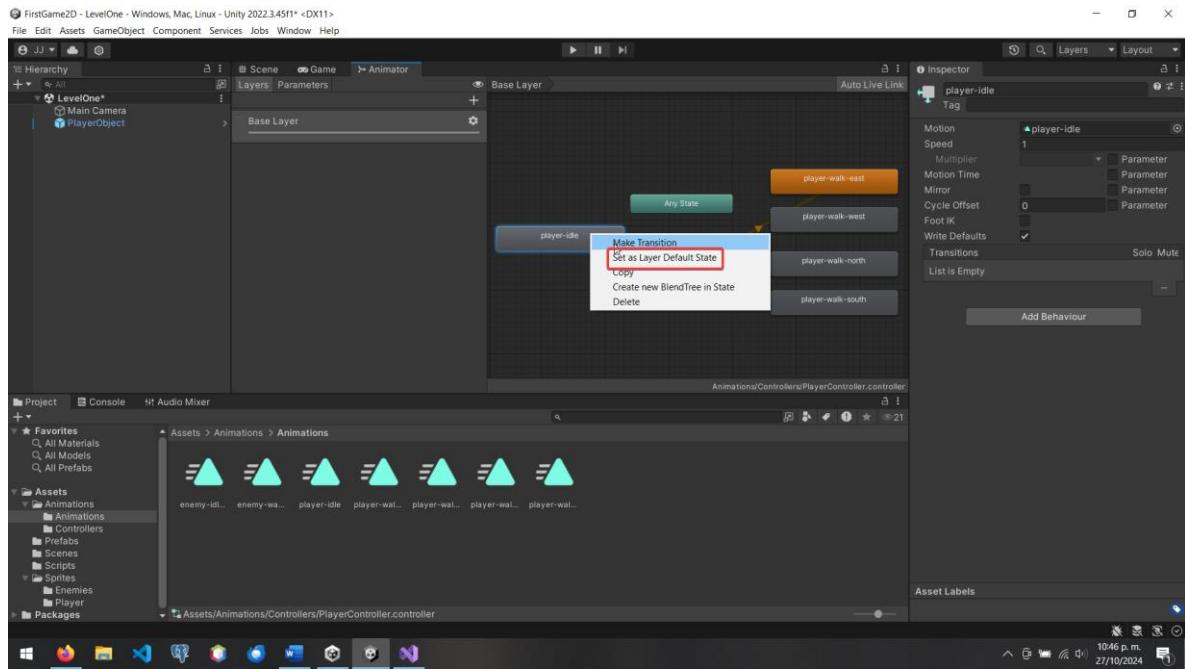
    // Start is called before the first frame update
    void Start()
    {
        rb2D = GetComponent<Rigidbody2D>();
    }

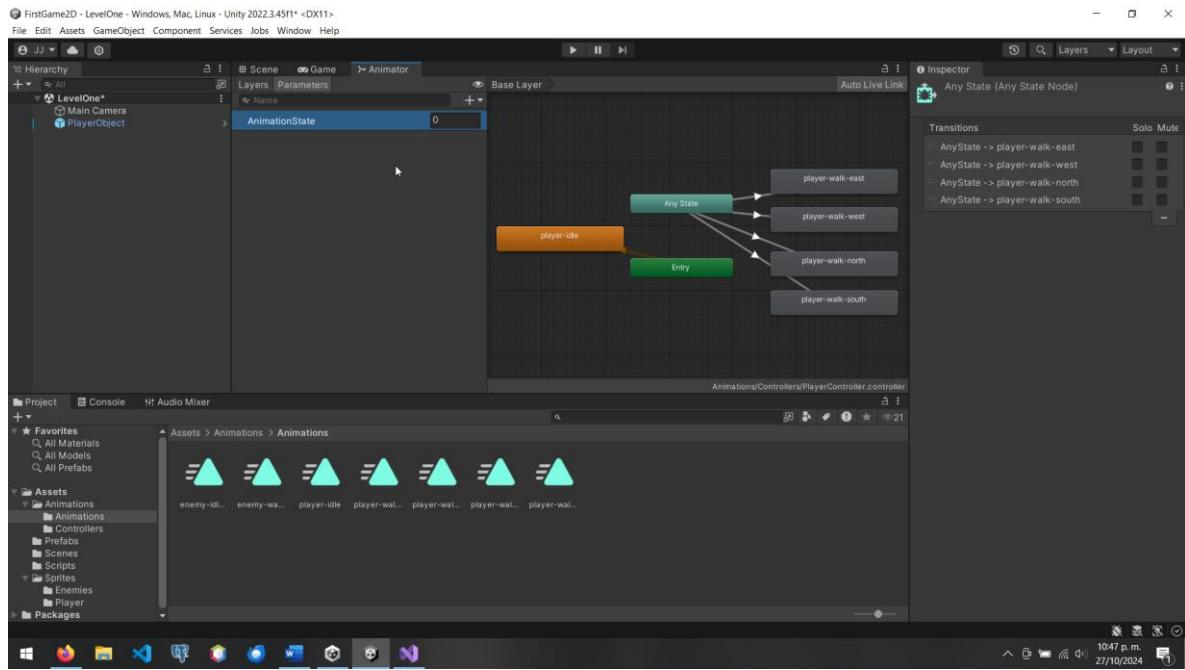
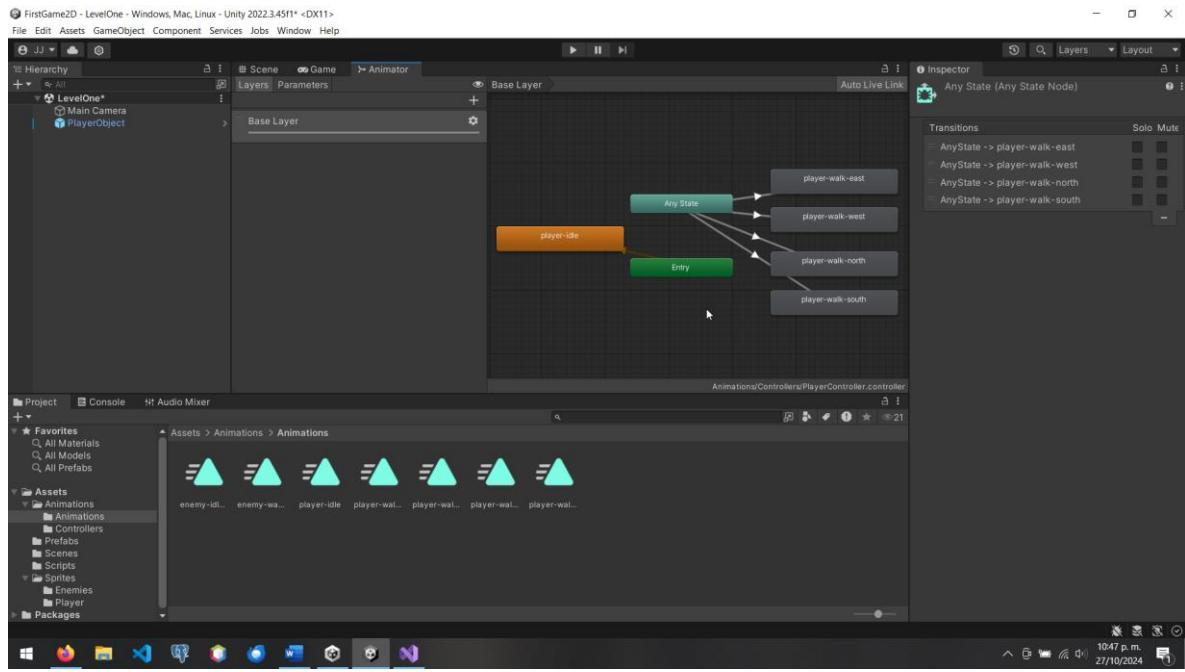
    // Update is called once per frame
    private void FixedUpdate()
    {
        // Captura los datos de entrada del usuario
        movement.x = Input.GetAxisRaw("Horizontal");
        movement.y = Input.GetAxisRaw("Vertical");

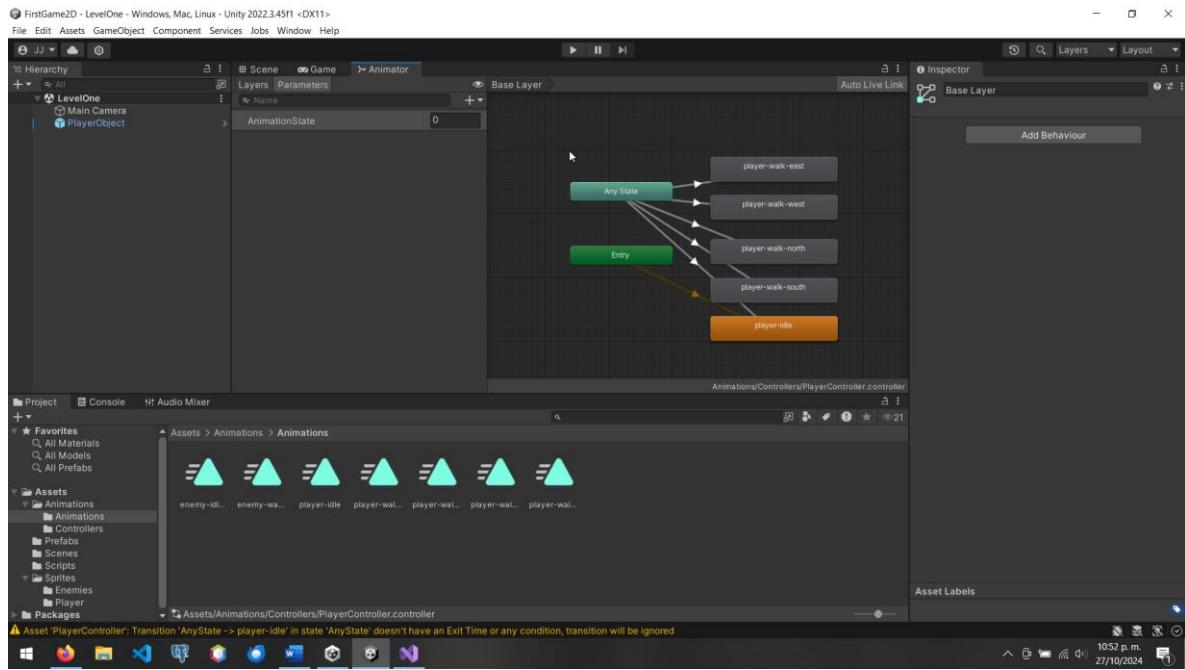
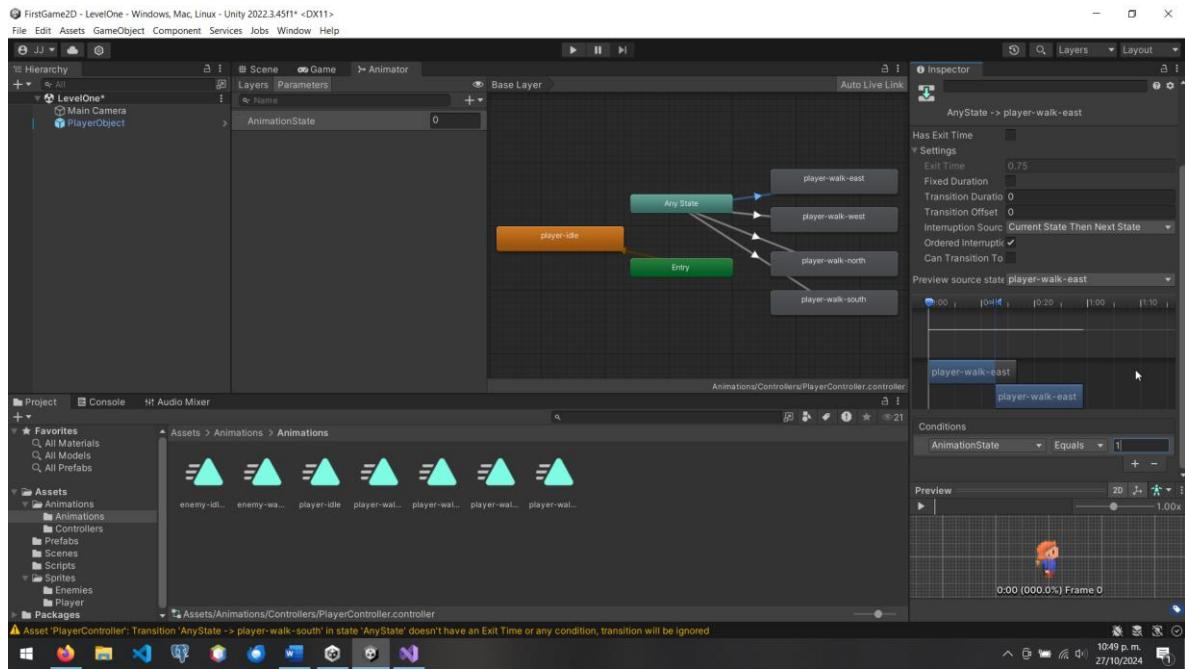
        // Conserva el rango de velocidad
        movement.Normalize();
        rb2D.velocity = movement * movementSpeed;
    }
}
```

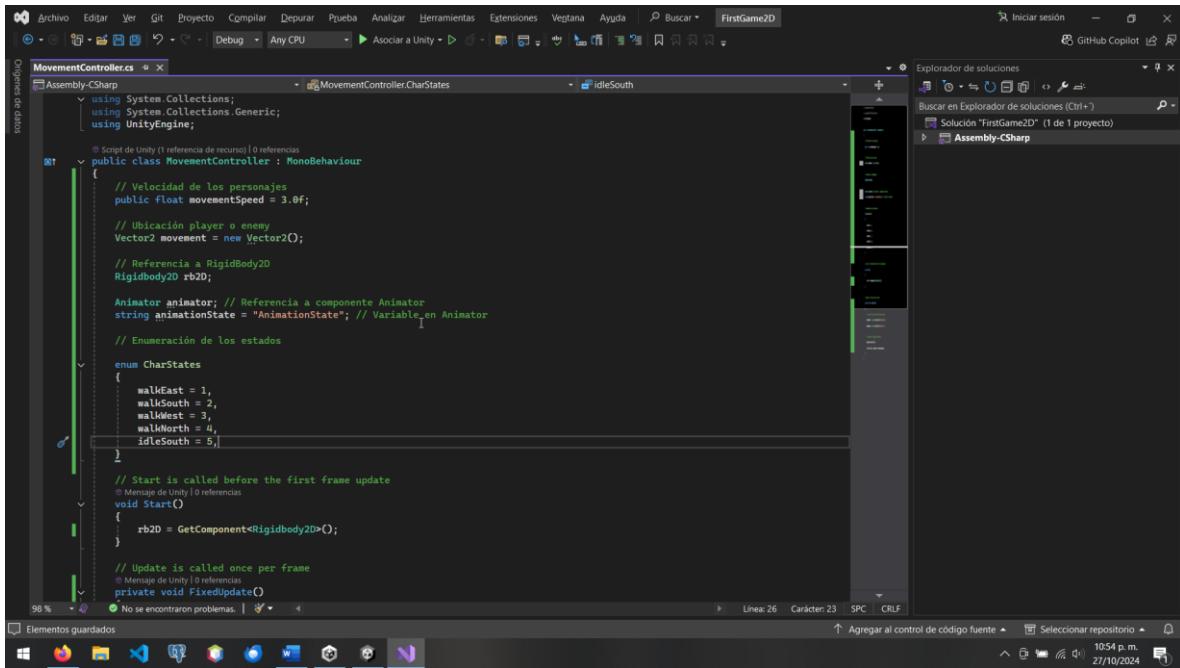












```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class MovementController : MonoBehaviour
{
    // Velocidad de los personajes
    public float movementSpeed = 3.0f;

    // Ubicación player o enemy
    Vector2 movement = new Vector2(0, 0);

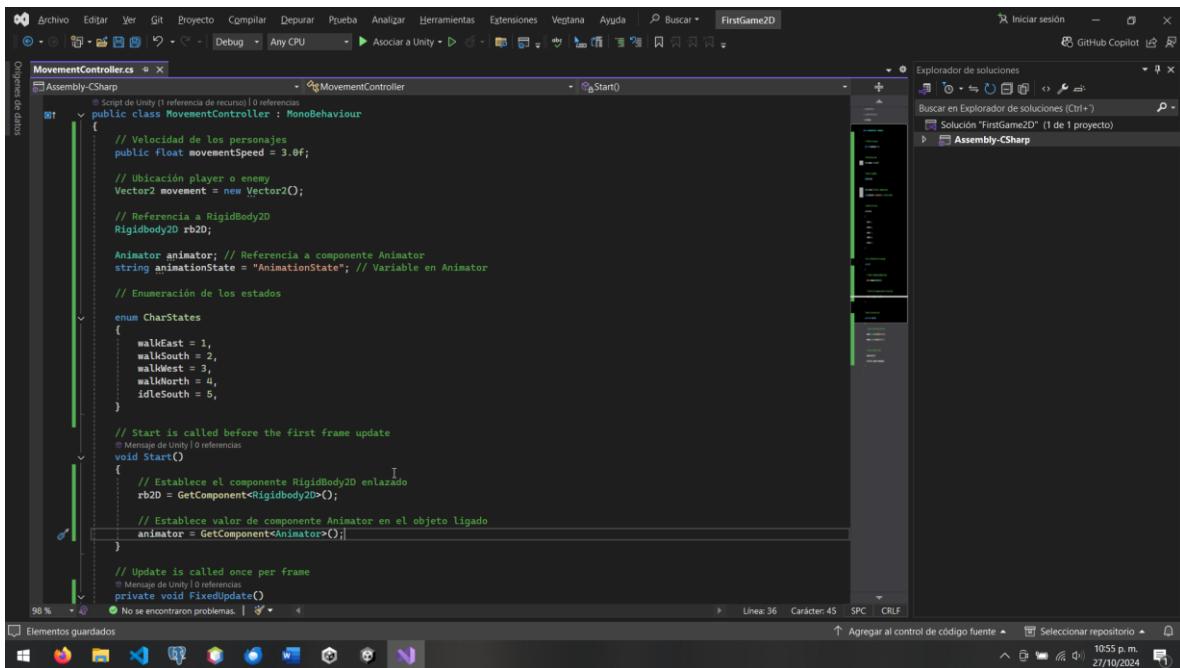
    // Referencia a Rigidbody2D
    Rigidbody2D rb2D;

    Animator animator; // Referencia a componente Animator
    string animationState = "AnimationState"; // Variable en Animator

    // Enumeración de los estados
    enum CharStates
    {
        walkEast = 1,
        walkSouth = 2,
        walkWest = 3,
        walkNorth = 4,
        idleSouth = 5,
    }

    // Start is called before the first frame update
    void Start()
    {
        rb2D = GetComponent<Rigidbody2D>();
    }

    // Update is called once per frame
    private void FixedUpdate()
    {
        // No se encontraron problemas.
    }
}
```



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

public class MovementController : MonoBehaviour
{
    // Velocidad de los personajes
    public float movementSpeed = 3.0f;

    // Ubicación player o enemy
    Vector2 movement = new Vector2(0, 0);

    // Referencia a Rigidbody2D
    Rigidbody2D rb2D;

    Animator animator; // Referencia a componente Animator
    string animationState = "Animationstate"; // Variable en Animator

    // Enumeración de los estados
    enum CharStates
    {
        walkEast = 1,
        walkSouth = 2,
        walkWest = 3,
        walkNorth = 4,
        idleSouth = 5,
    }

    // Start is called before the first frame update
    void Start()
    {
        // Establece el componente Rigidbody2D enlazado
        rb2D = GetComponent<Rigidbody2D>();

        // Establece valor de componente Animator en el objeto ligado
        animator = GetComponent<Animator>();
    }

    // Update is called once per frame
    private void FixedUpdate()
    {
        // No se encontraron problemas.
    }
}
```

Screenshot of Visual Studio showing the MovementController.cs code. The code defines a class MovementController that interacts with an Animator component to set movement states based on user input.

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

public class MovementController : MonoBehaviour
{
    // Establece valor de componente Animator en el objeto ligado
    animator = GetComponent<Animator>();

    // Update es llamado una vez por frame
    // Mensaje de Unity | 0 referencias
    private void Update()
    {
        this.UpdateState();
    }

    /*
     * Método que define la definición a ejecutar en base al movimiento realizado por el usuario.
     */
    // Referencia
    private void UpdateState()
    {
        if (movement.x > 0)
        { //ESTE
            anim.(Campo)Animator MovementController.animator rStates.walkEast;
        }
        else if (movement.x < 0)
        { //OESTE
            animator.SetInteger(animationState, (int)CharStates.walkWest);
        }
        else if (movement.y > 0)
        { //NORTE
            animator.SetInteger(animationState, (int)CharStates.walkNorth);
        }
        else if (movement.y < 0)
        { //SUR
            animator.SetInteger(animationState, (int)CharStates.walkSouth);
        }
        else
        { //IDLE
            animator.SetInteger(animationState, (int)CharStates.idleSouth);
        }
    }

    // Mensaje de Unity | 0 referencias
    private void FixedUpdate()
    {
        MoveCharacter();
    }

    // Referencia
    private void MoveCharacter()
    {
        // Captura los datos de entrada del usuario
        movement.x = Input.GetAxisRaw("Horizontal");
        movement.y = Input.GetAxisRaw("Vertical");

        // Conserva el rango de velocidad
        movement.Normalize();
        rb2D.velocity = movement * movementSpeed;
    }
}
```

Screenshot of Visual Studio showing the MovementController.cs code with additional logic added. The code now includes a FixedUpdate() method and a MoveCharacter() method to handle user input and apply movement.

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

public class MovementController : MonoBehaviour
{
    // Establece valor de componente Animator en el objeto ligado
    animator = GetComponent<Animator>();

    // Update es llamado una vez por frame
    // Mensaje de Unity | 0 referencias
    private void Update()
    {
        this.UpdateState();
    }

    /*
     * Método que define la definición a ejecutar en base al movimiento realizado por el usuario.
     */
    // Referencia
    private void UpdateState()
    {
        if (movement.x > 0)
        { //ESTE
            animator.SetInteger(animationState, (int)CharStates.walkEast);
        }
        else if (movement.y > 0)
        { //NORTE
            animator.SetInteger(animationState, (int)CharStates.walkNorth);
        }
        else if (movement.y < 0)
        { //SUR
            animator.SetInteger(animationState, (int)CharStates.walkSouth);
        }
        else
        { //IDLE
            animator.SetInteger(animationState, (int)CharStates.idleSouth);
        }
    }

    // Mensaje de Unity | 0 referencias
    private void FixedUpdate()
    {
        MoveCharacter();
    }

    // Referencia
    private void MoveCharacter()
    {
        // Captura los datos de entrada del usuario
        movement.x = Input.GetAxisRaw("Horizontal");
        movement.y = Input.GetAxisRaw("Vertical");

        // Conserva el rango de velocidad
        movement.Normalize();
        rb2D.velocity = movement * movementSpeed;
    }
}
```

