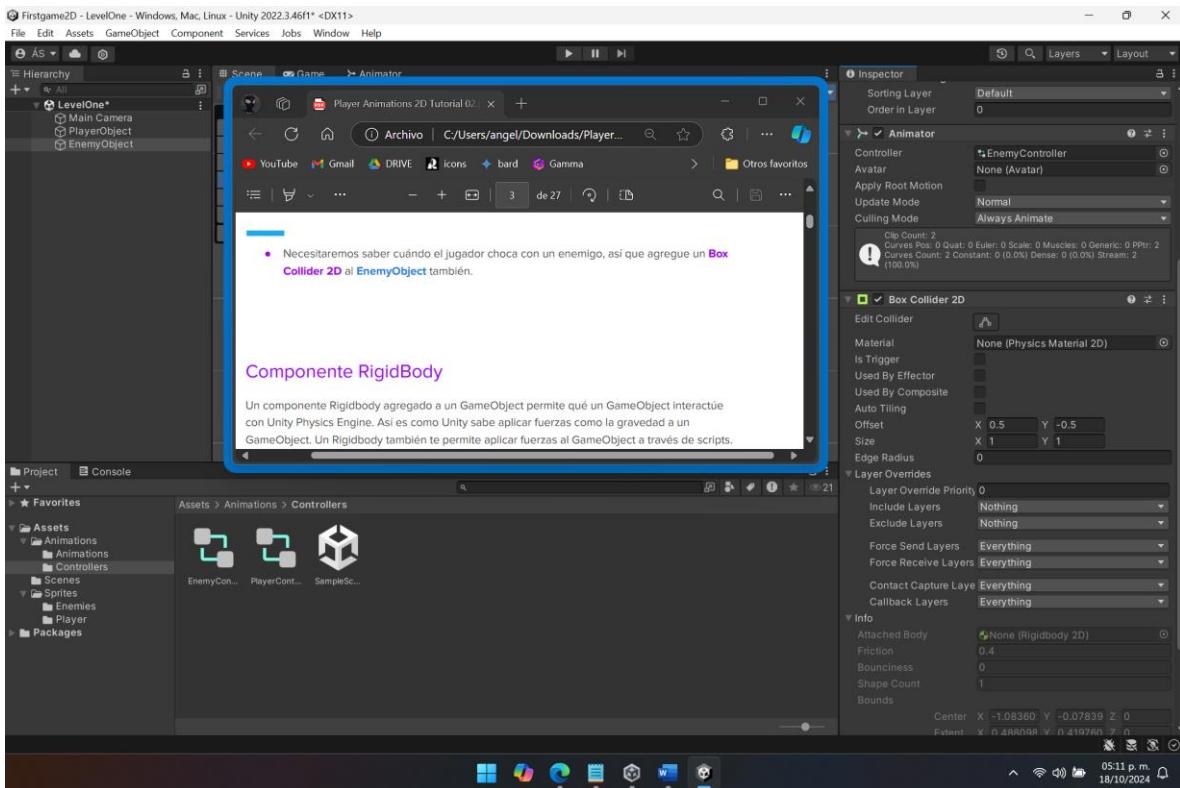
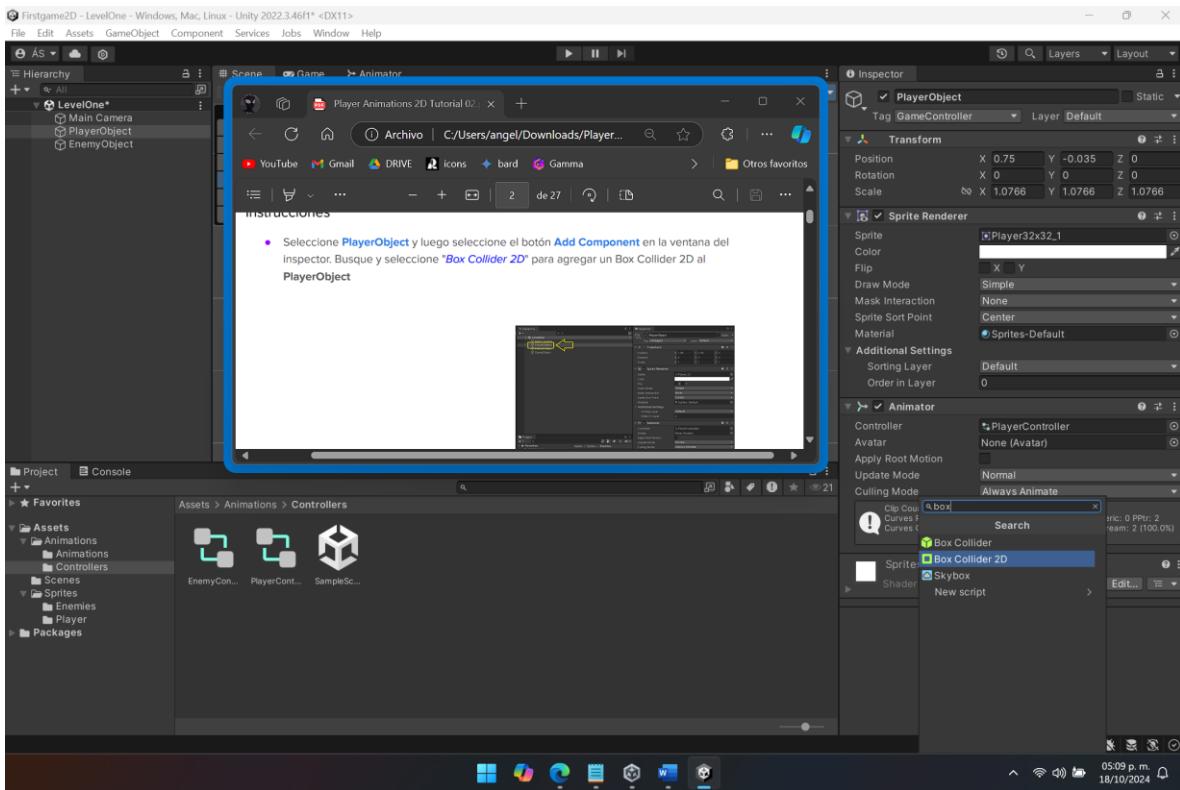
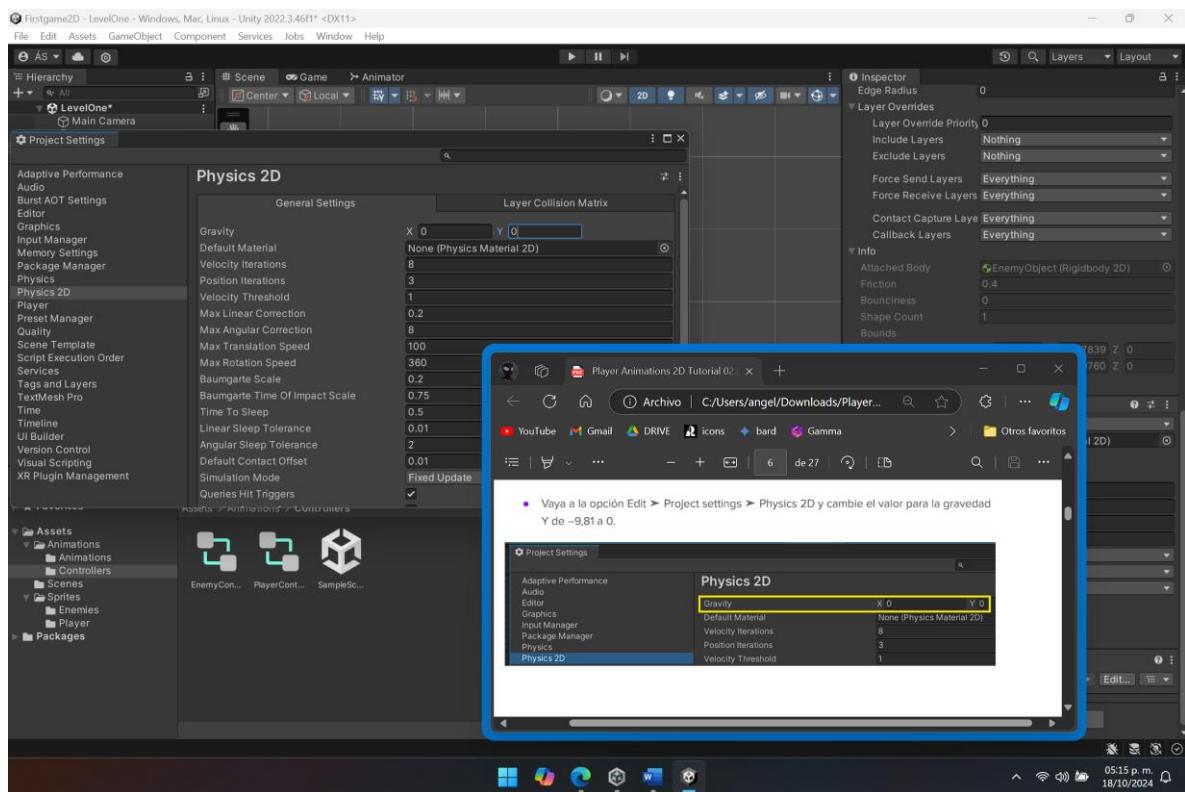
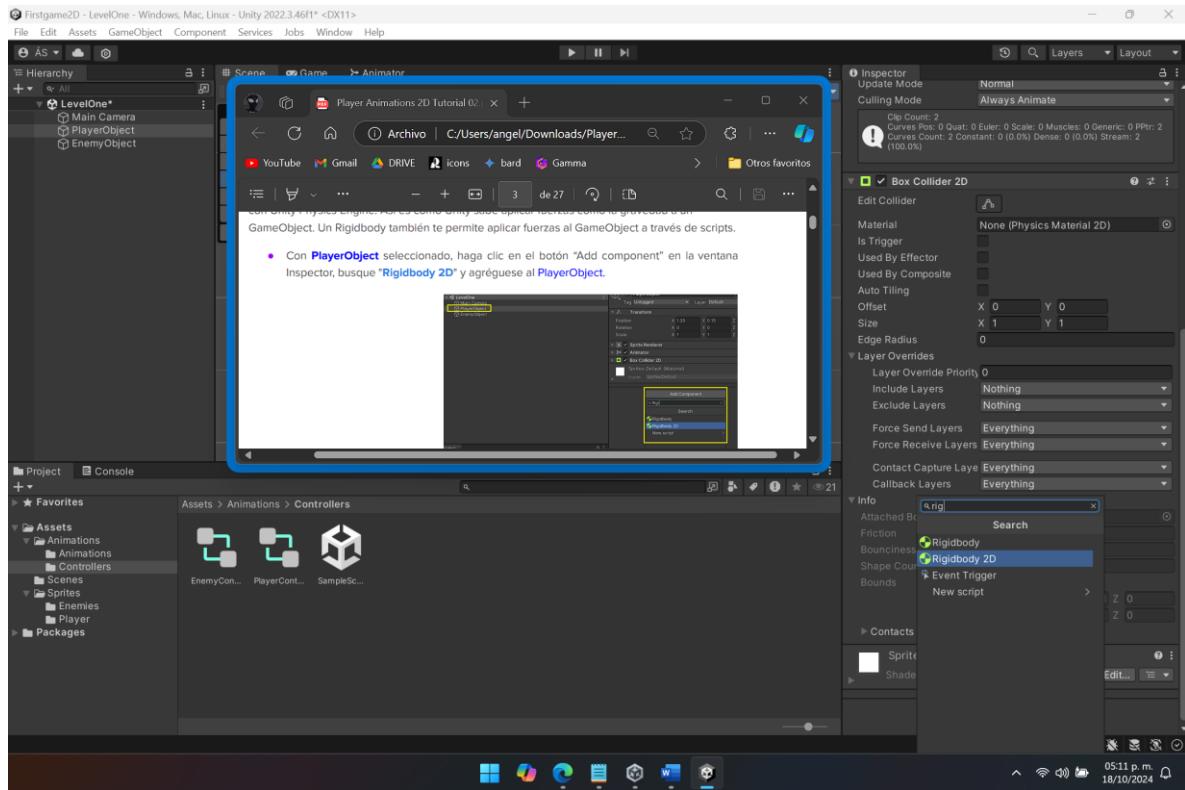
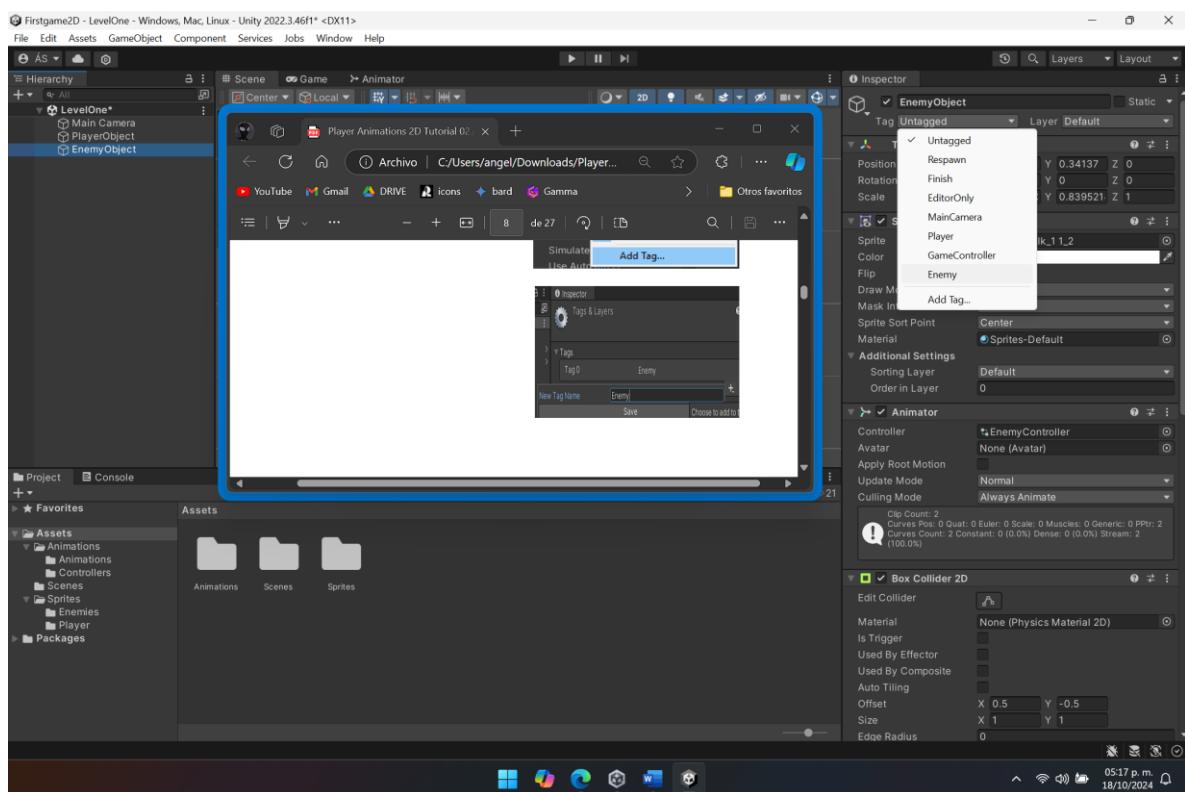
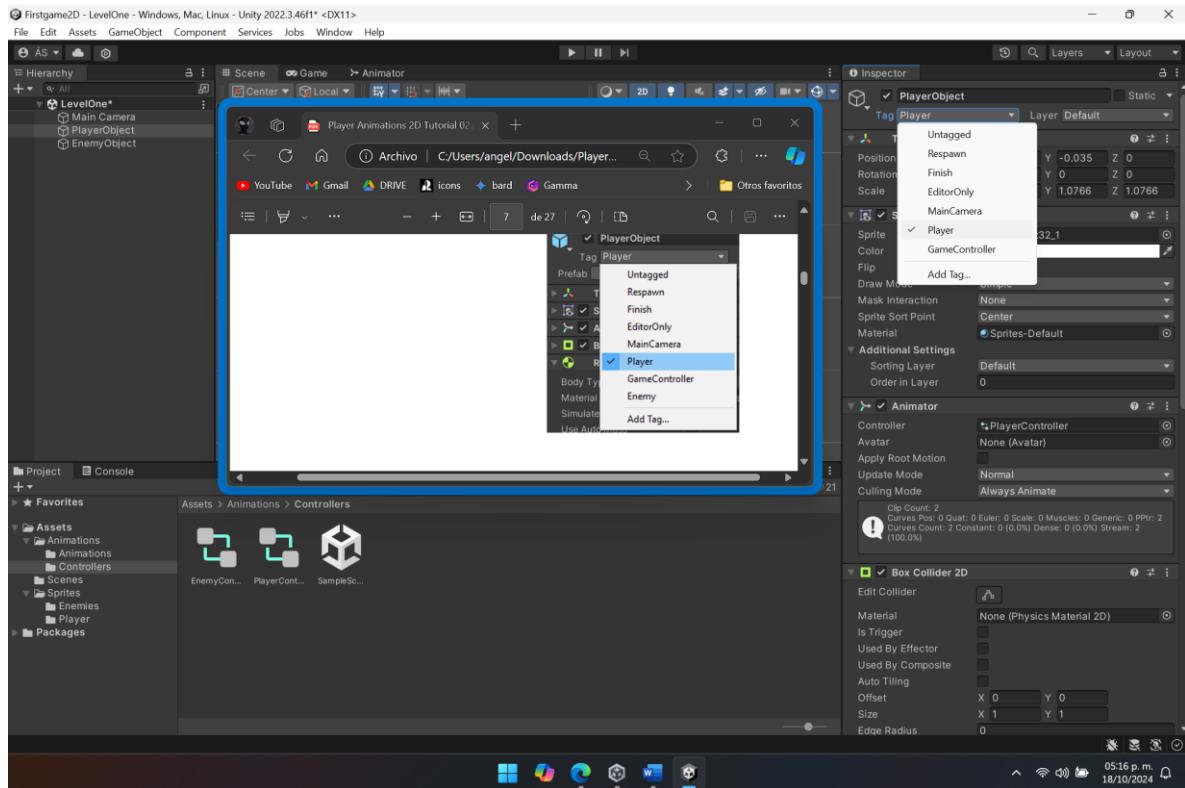


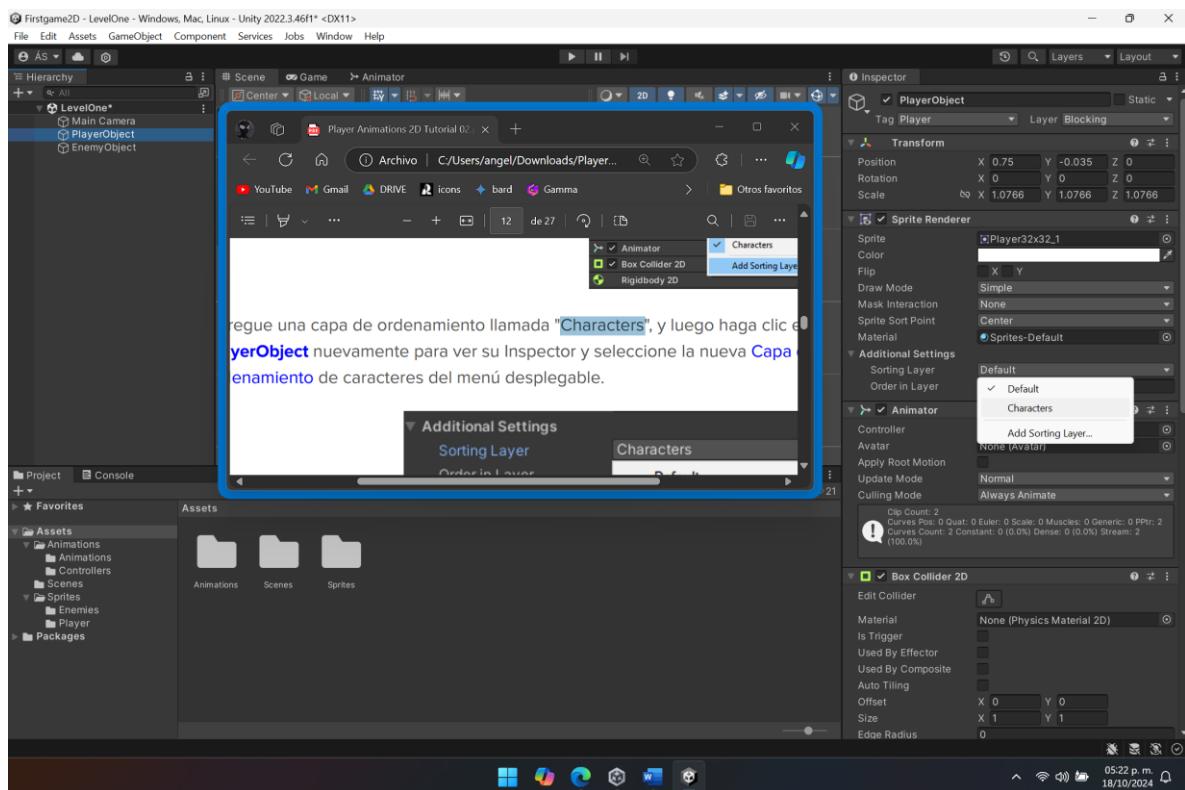
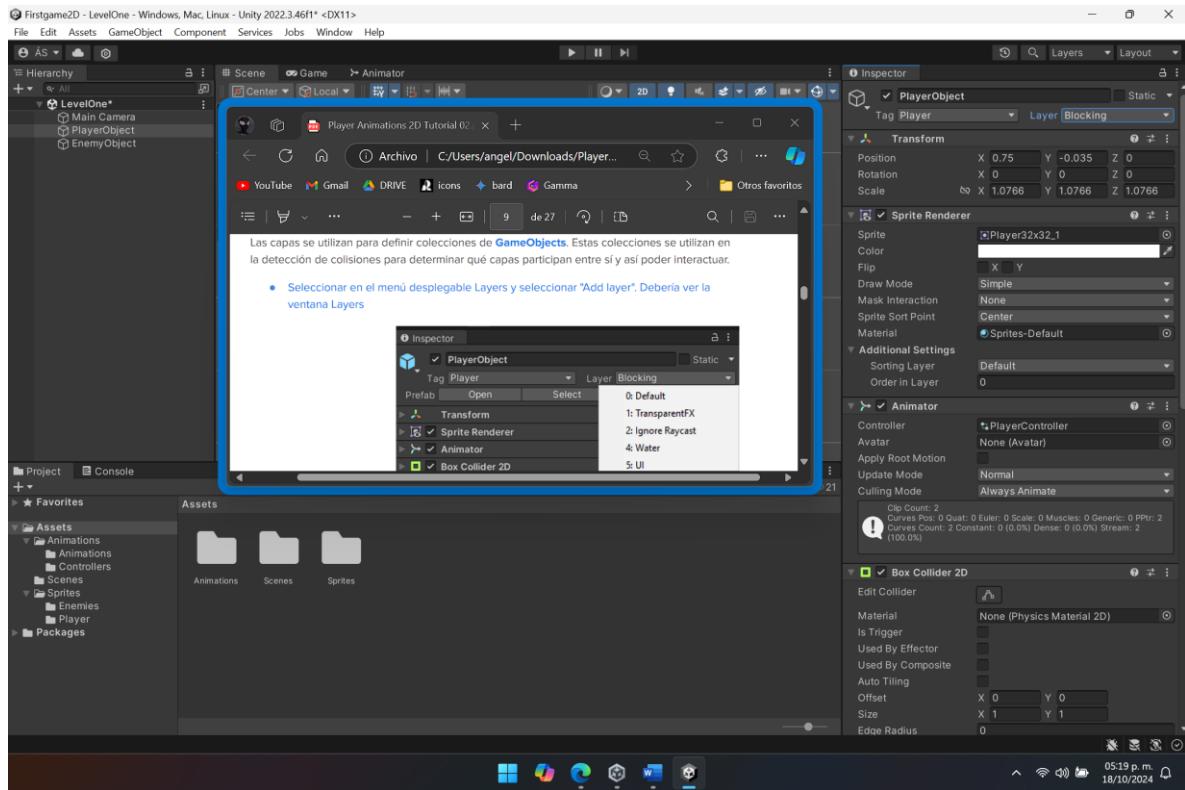
1

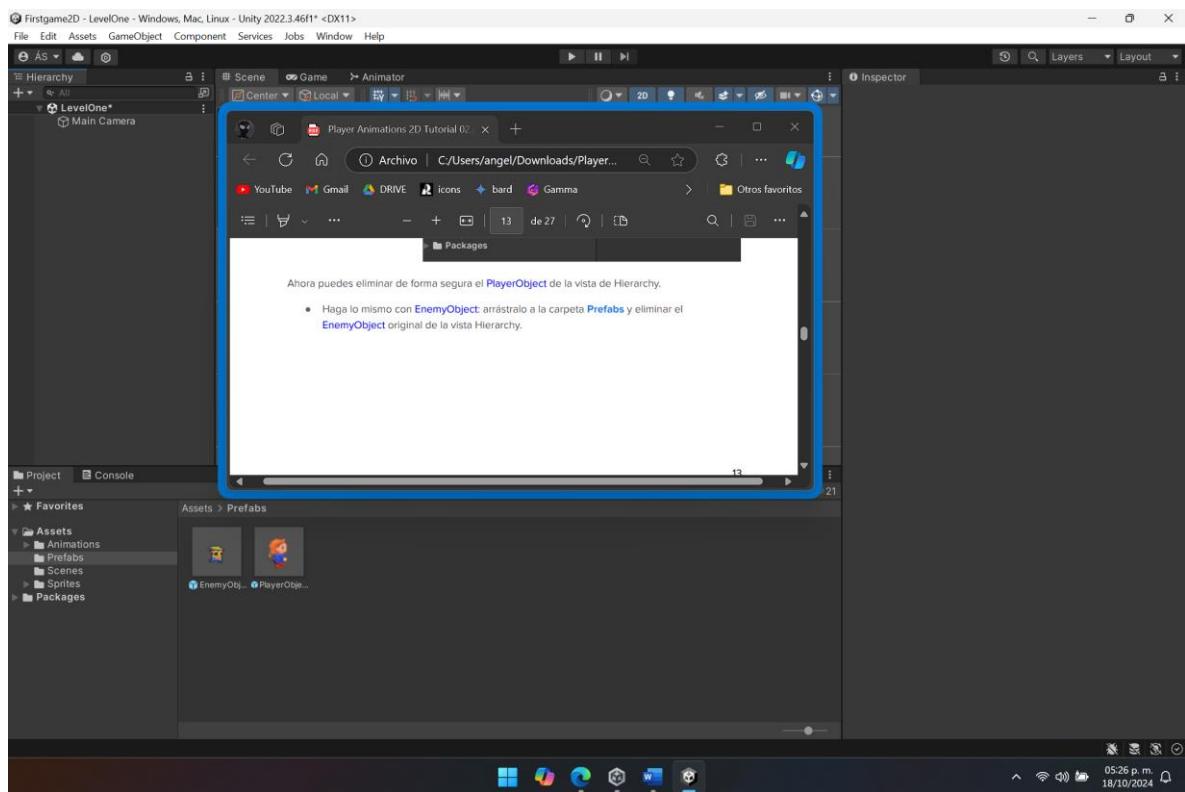
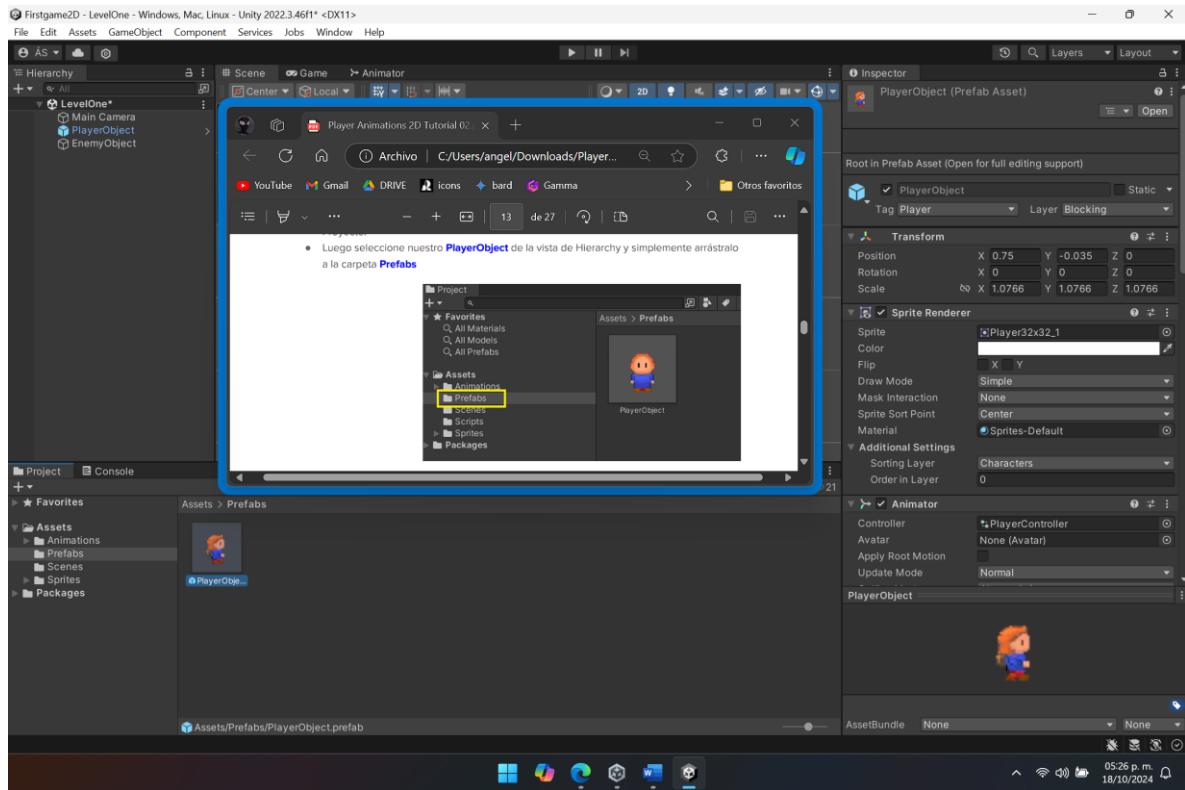
2

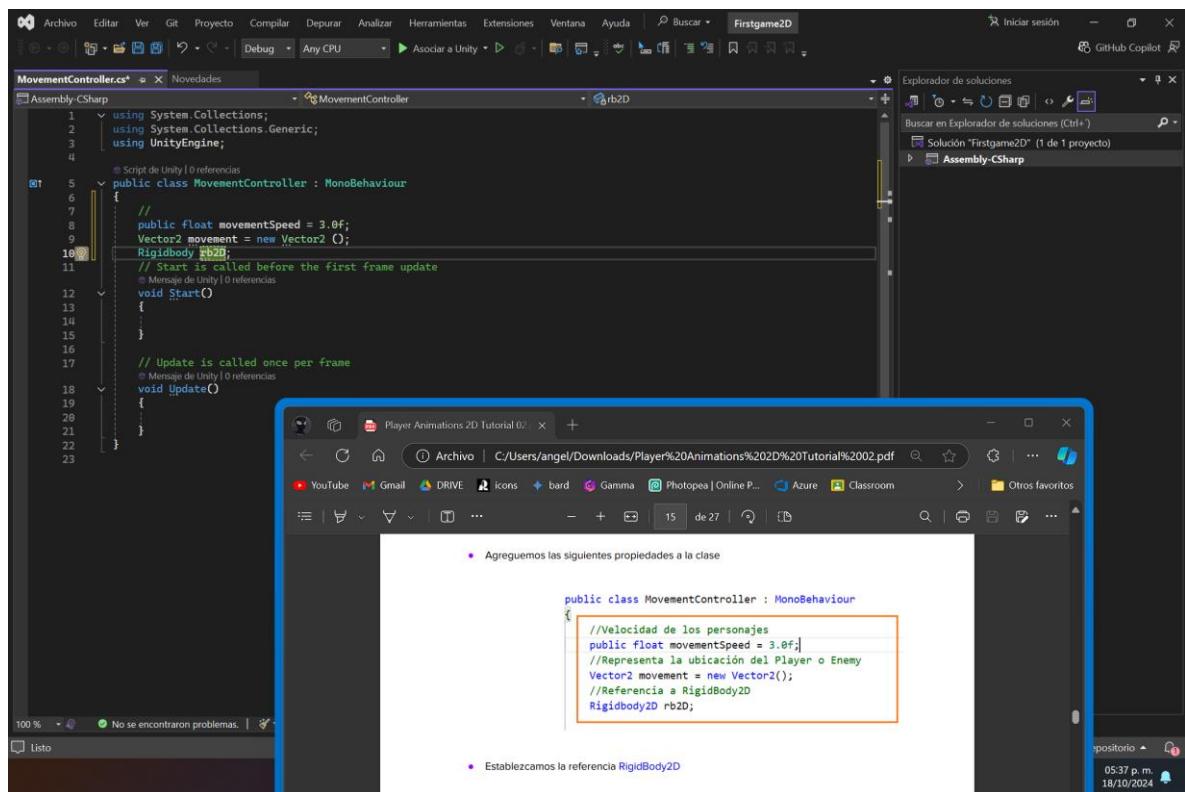
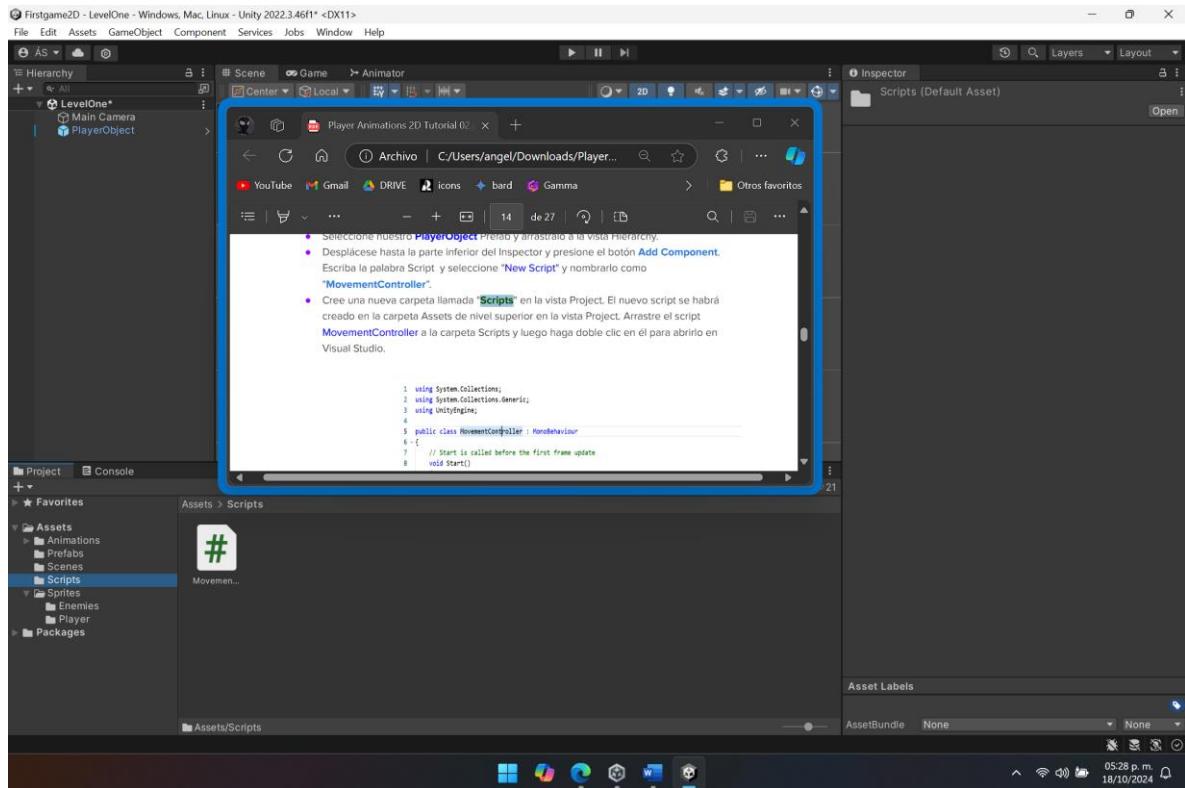












The screenshot shows the Unity Editor's code editor with the file `MovementController.cs` open. The code defines a `MovementController` class that inherits from `MonoBehaviour` . It includes methods for `Start()`, `Update()`, and `FixedUpdate()`. The `Update()` method uses `Input.GetAxis` to get horizontal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

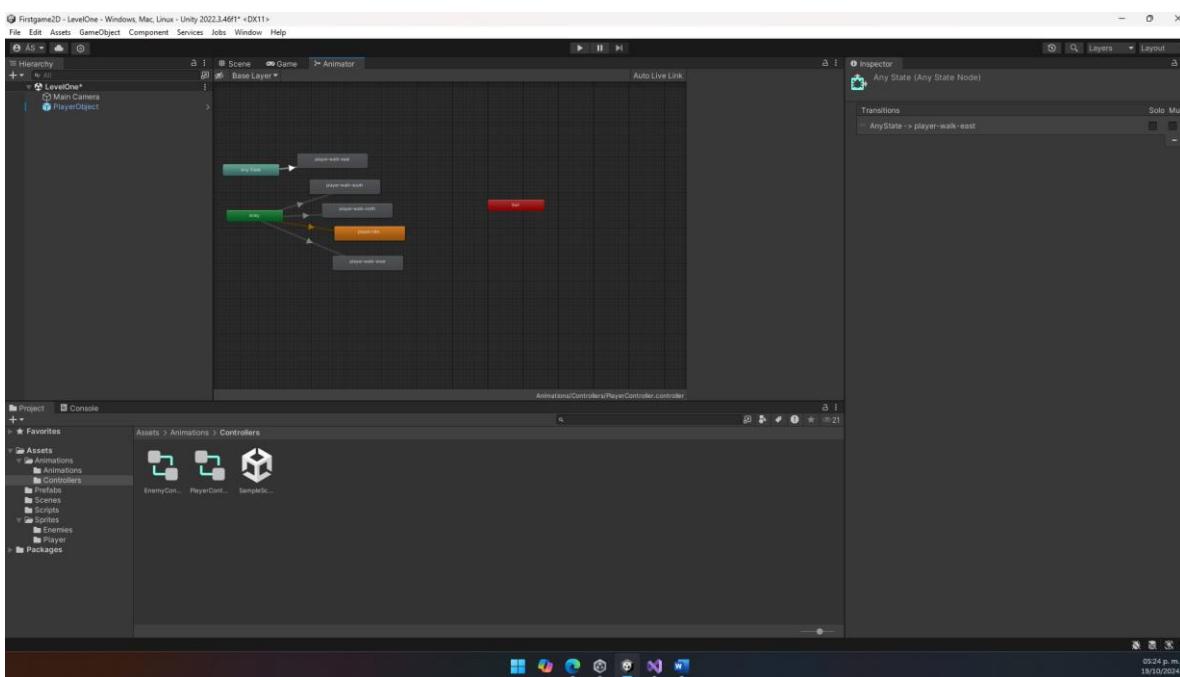
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

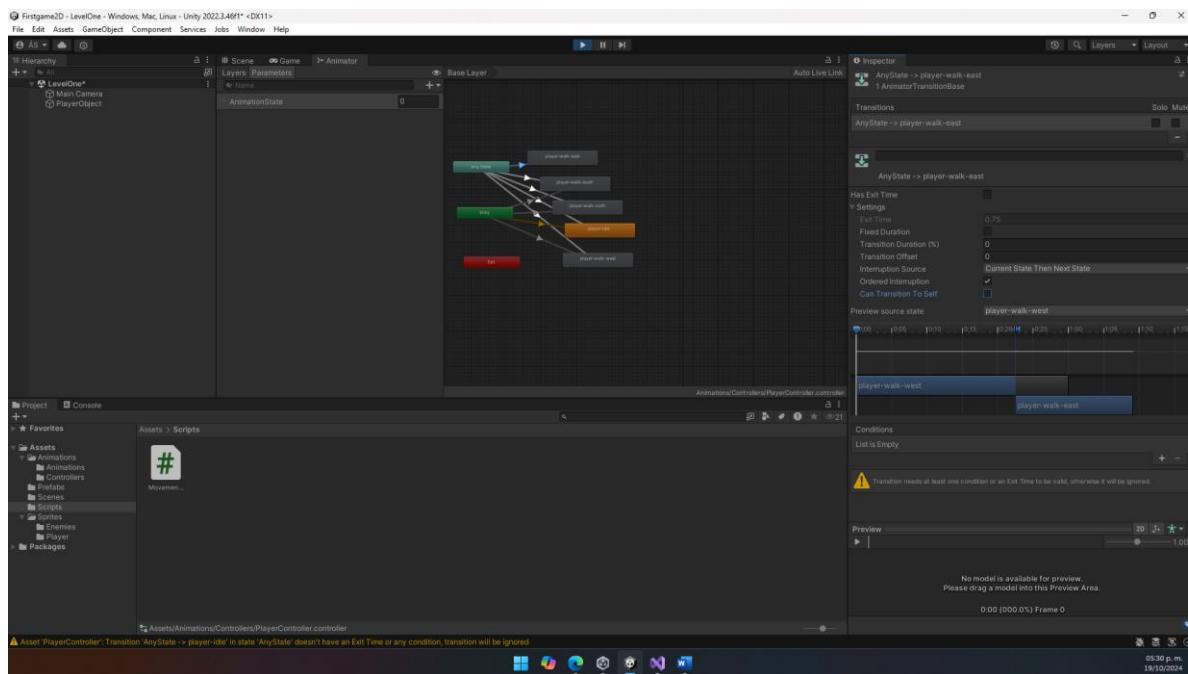
public class MovementController : MonoBehaviour
{
    public float movementSpeed = 3.0f;
    Vector2 movement = new Vector2(0, 0);

    // 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 with the script `MovementController.cs` open. The code defines a `MovementController` class that extends `MonoBehaviour` . It includes fields for movement speed, movement vector, and rigidbody. An animator component is referenced. The script uses an enum `CharStates` with five values: `walkEast`, `walkSouth`, `walkWest`, `walkNorth`, and `idleSouth`. The `Start()` method initializes the rigidbody and animator. The `Update()` method calls `UpdateState()`. The `UpdateState()` method logic handles movement direction based on the movement vector (`x > 0` for east, `x < 0` for west, `y > 0` for north, `y < 0` for south) and sets the appropriate animation state using the `SetInteger` method.

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

public class MovementController : MonoBehaviour
{
    public float movementSpeed = 2.0f;
    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< Rigidbody2D>();

        //Referencia al componente Animator el objeto ligado
        animator = GetComponent< Animator>();
    }

    // Update is called once per frame
    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)
        {
            animator.SetInteger(animationState, (int)CharStates.walkNorth);
        }
        else if (movement.y < 0)
        {
            animator.SetInteger(animationState, (int)CharStates.idleSouth);
        }
    }
}
```

The screenshot shows the Unity Editor's code editor with the file `MovementController.cs` open. The code is written in C# and defines a class `MovementController` with methods `update`, `FixedUpdate`, and `MoveCharacter`. The `update` method calls `UpdateState` and `MoveCharacter`. The `UpdateState` method sets the animation state based on movement direction. The `MoveCharacter` method gets input axes, normalizes them, and applies movement speed to a rigidbody2D.

```
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)
        {
            animator.SetInteger(animationState, (int)CharStates.walkNorth);
        }
        else if (movement.y < 0)
        {
            animator.SetInteger(animationState, (int)CharStates.walkSouth);
        }
        else
        {
            animator.SetInteger(animationState, (int)CharStates.idleSouth);
        }
    }

    void FixedUpdate()
    {
        MoveCharacter(); //Metodo definido para ingresar la direccion del personaje
    }

    private void MoveCharacter()
    {
        movement.x = Input.GetAxis("Horizontal");
        movement.y = Input.GetAxis("Vertical");

        movement.Normalize();
        rb2D.velocity = movement * movementSpeed;
    }
}
```