Elaborado por el docente Gabriel Barrón Rodríguez

Programación de Videojuegos **Player Animations 2D**

Segunda parte

Objetivo

- ★ Usaremos los componentes Colliders y RigidBody para la física de los objetos.
- ★ Utilizaremos Sorting Layers para el ordenamiento de los objetos.
- ★ Reutilizamiento de objetos con Prefabs.

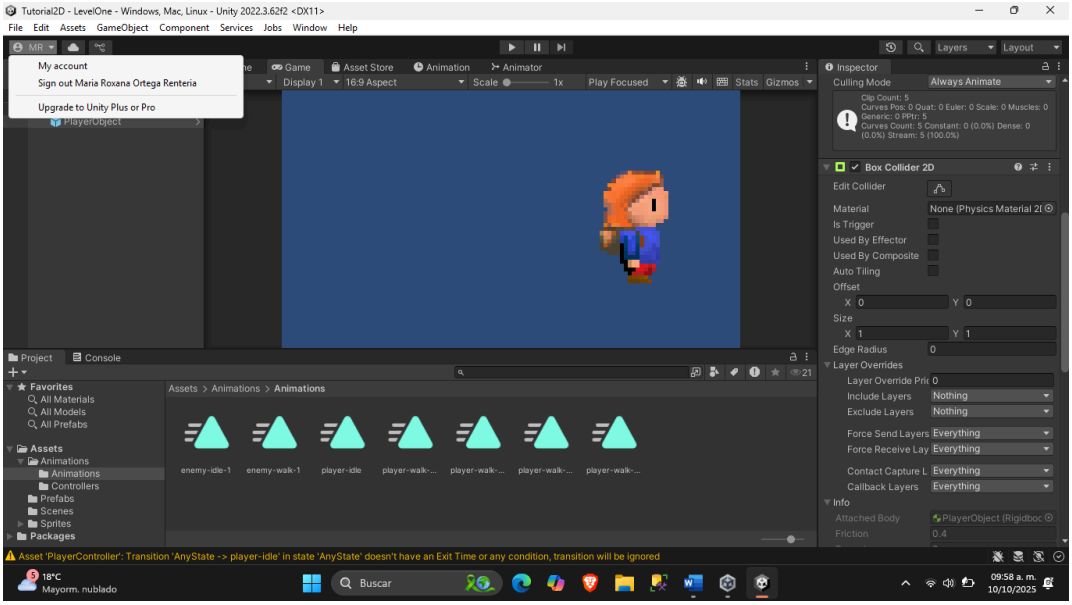
Colliders

Los colisionadores se agregan a GameObjects y son utilizados por Unity Physics Engine para determinar cuándo se ha producido una colisión entre dos objetos. La forma de un colisionador es ajustable, y por lo general tienen una forma más o menos parecida al contorno del objeto que representan.

Una aproximación de la forma de los objetos es usando un tipo de colisionador llamado "Primitive Collider" es menos intenso para el procesador. Hay dos tipos de colisionadores primitivos en Unity 2D: **Box Collider 2D** y **Circle Collider 2D**.

Instrucciones

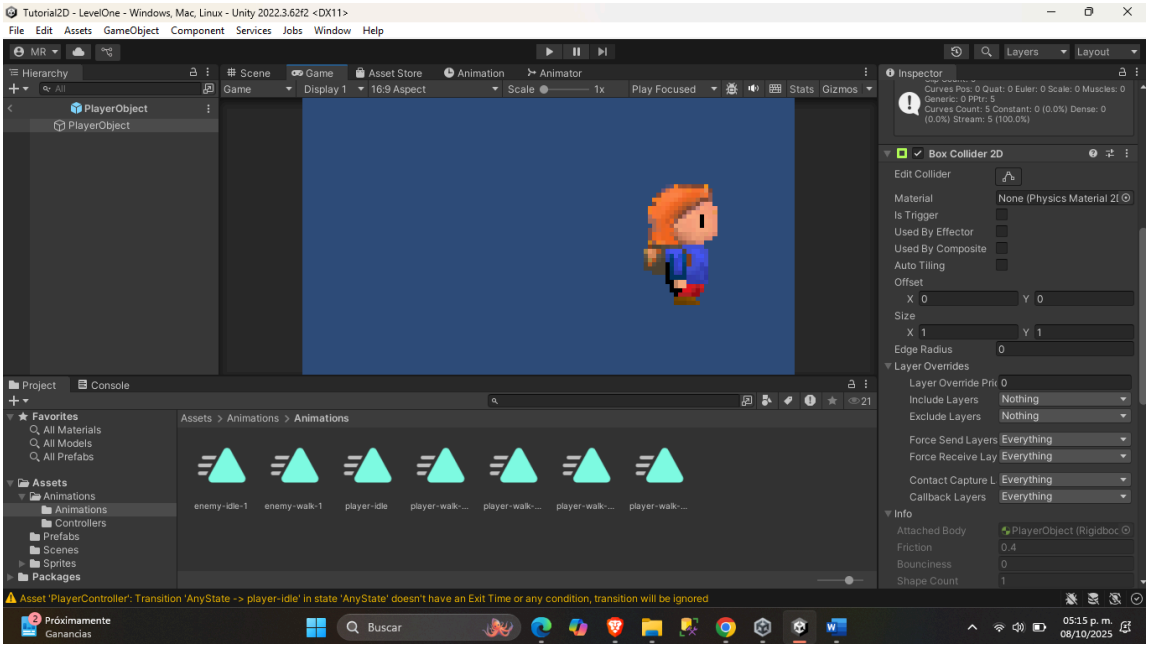
- Seleccione **PlayerObject** y luego seleccione el botón **Add Component** en la ventana del inspector. Busque y seleccione "*Box Collider 2D*" para agregar un Box Collider 2D al **PlayerObject**
- Necesitaremos saber cuándo el jugador choca con un enemigo, así que agregue un **Box Collider 2D** al **EnemyObject** también.



Componente RigidBody

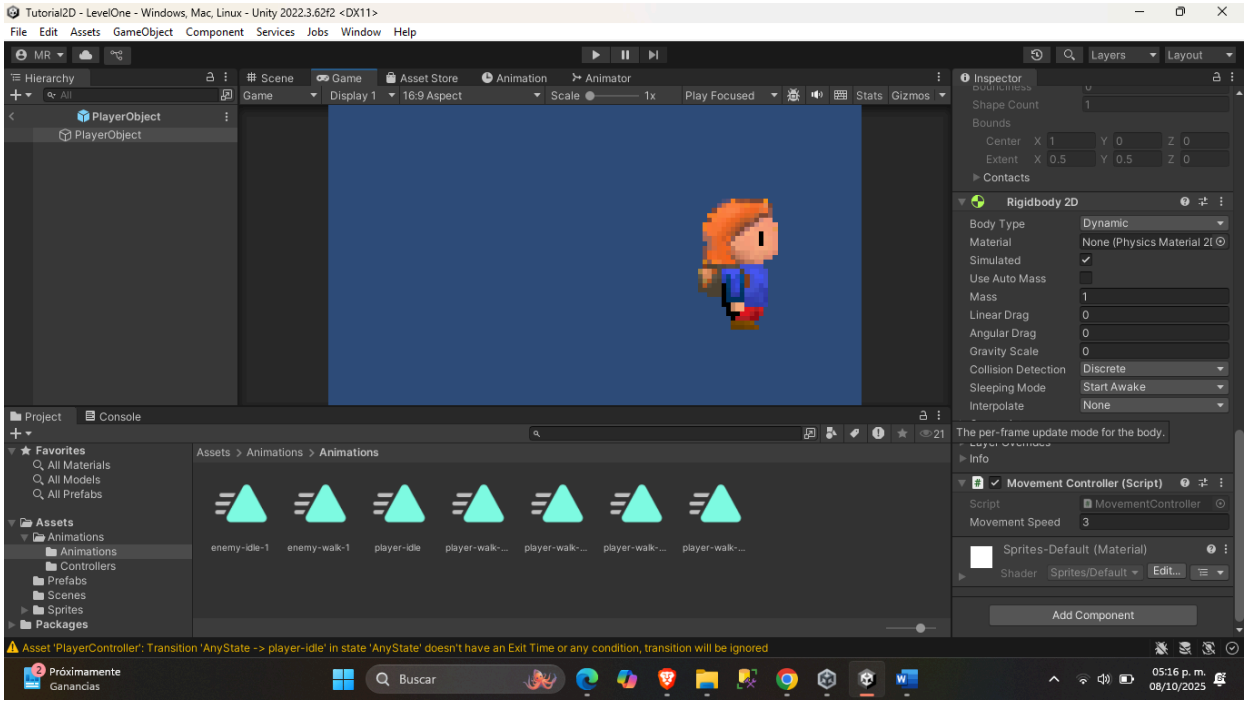
Un componente Rigidbody agregado a un GameObject permite qué un GameObject interactúe con Unity Physics Engine. Así es como Unity sabe aplicar fuerzas como la gravedad a un GameObject. Un Rigidbody también te permite aplicar fuerzas al GameObject a través de scripts.

- Con **PlayerObject** seleccionado, haga clic en el botón “Add component” en la ventana Inspector, busque "**Rigidbody 2D**" y agréguese al PlayerObject.

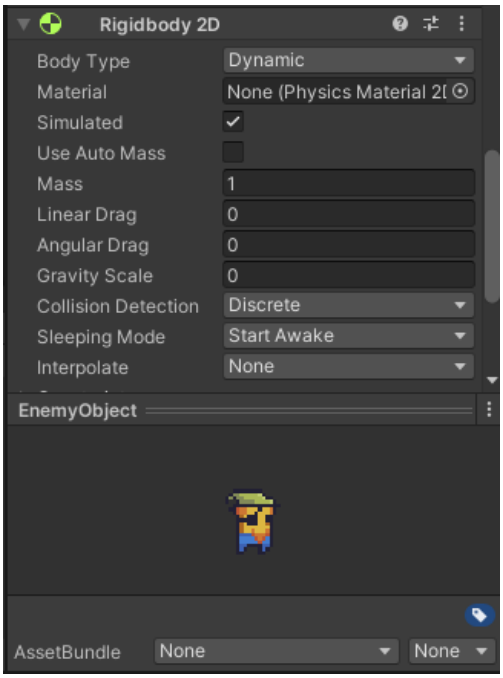


En el menú desplegable establezca los valores a las siguientes propiedades:

Body Type	Dynamic
Mass	1
Linear Drag	0
Angular Drag	0
Gravity Scale	0

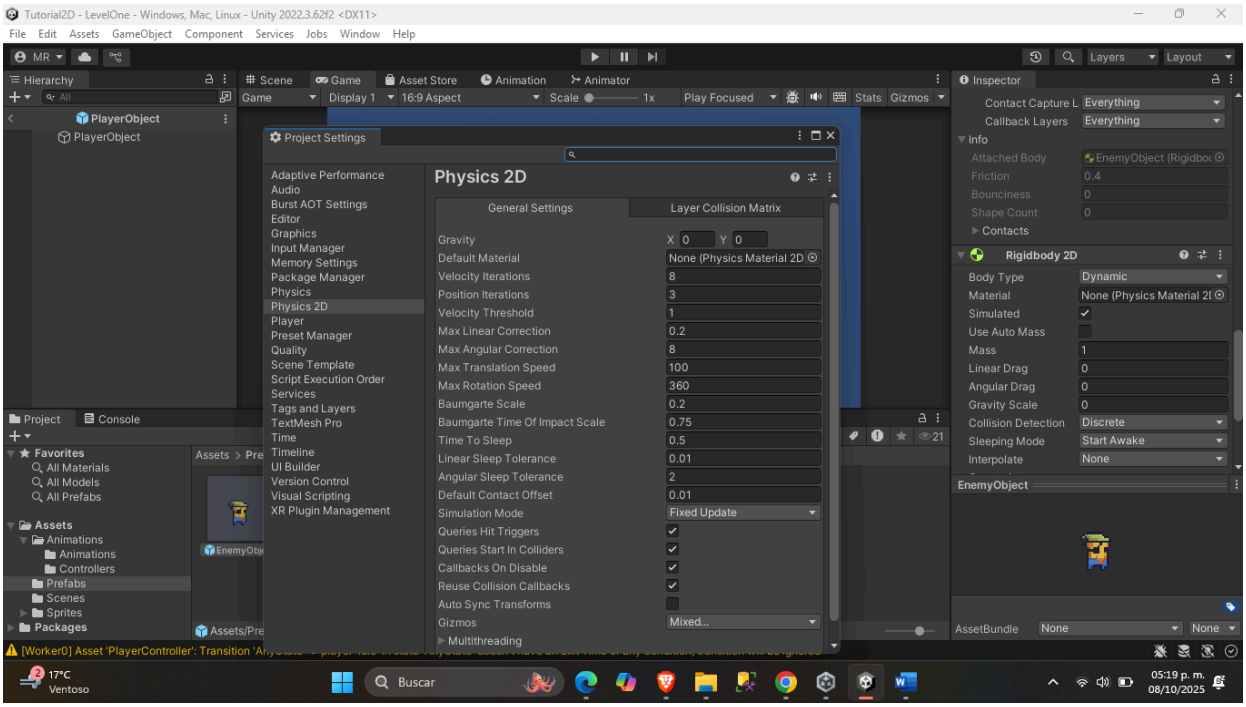


Seleccione el **EnemyObject** y agregue un Componente Rigidbody 2D de tipo Dinámico para él también.



Ahora que agregamos un Rigidbody 2D a nuestro jugador y enemigo, se verán afectados por la gravedad. Debido a que nuestro juego usa una perspectiva top-down, apaguemos la gravedad para que nuestro jugador no salga volando en la pantalla.

Vaya a la opción Edit ➤ Project settings ➤ Physics 2D y cambie el valor para la gravedad Y de –9,81 a 0.

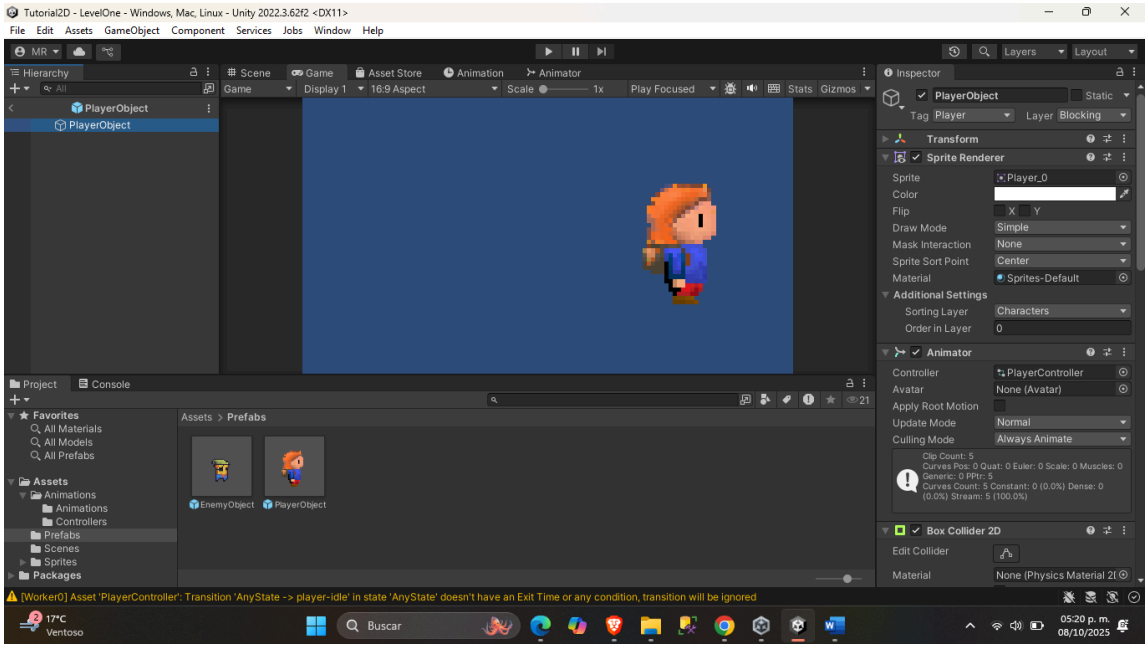

Tutorial2D - LevelOne - Windows, Mac, Linux - Unity 2022.3.62f2 <DX11>
File Edit Assets GameObject Component Services Jobs Window Help
Hierarchy
PlayerObject
Scene Game Asset Store Animation Animator
Project Settings
Physics 2D
General Settings
Layer Collision Matrix
Gravity
Default Material
Velocity Iterations
Position Iterations
Velocity Threshold
Max Linear Correction
Max Angular Correction
Max Translation Speed
Max Rotation Speed
Baumgarte Scale
Baumgarte Time Of Impact Scale
Time To Sleep
Linear Sleep Tolerance
Angular Sleep Tolerance
Default Contact Offset
Simulation Mode
Queries Hit Triggers
Queries Start In Colliders
Callbacks On Disable
Reuse Collision Callbacks
Auto Sync Transforms
Gizmos
Multithreading
Inspector
Rigidbody 2D
Body Type
Material
Simulated
Use Auto Mass
Mass
Linear Drag
Angular Drag
Gravity Scale
Collision Detection
Sleeping Mode
Interpolate
EnemyObject
Project
Console
Favorites
Assets
Prefabs
Scenes
Sprites
Packages
[Worker0] Asset 'PlayerController': Transition 'Al
Buscar
17°C
Ventoso
05:19 p. m.
08/10/2025

Etiquetas & Capas

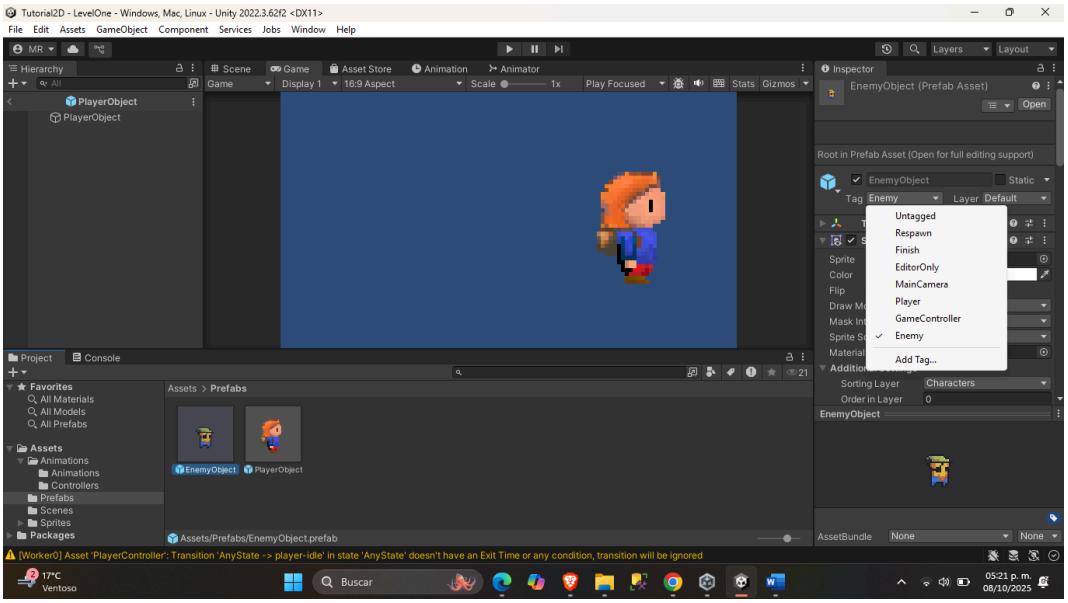
Etiquetas

Las etiquetas nos permiten etiquetar GameObjects para facilitar la referencia y la comparación mientras nuestro juego se está ejecutando.

- Selecciona el **PlayerObject**. En el menú desplegable Tag en la parte superior izquierda del Inspector, seleccione la etiqueta de Jugador para agregar una etiqueta a nuestro PlayerObject.



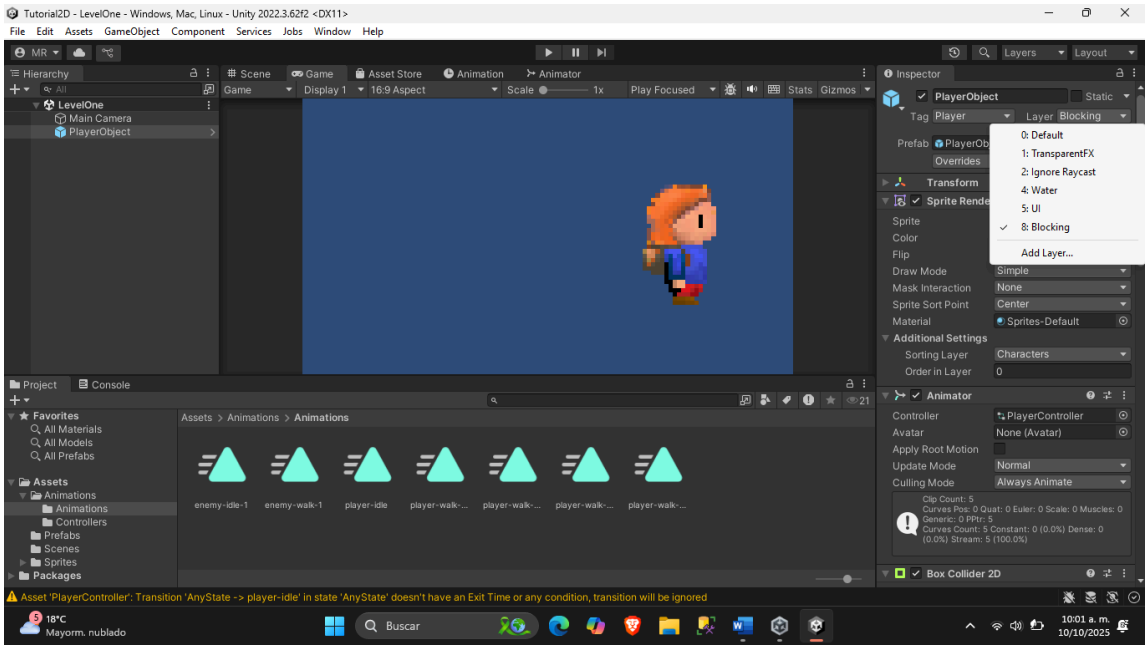
- Crear una nueva etiqueta llamada "Enemy" y úsala para configurar la etiqueta del objeto **EnemyObject**.



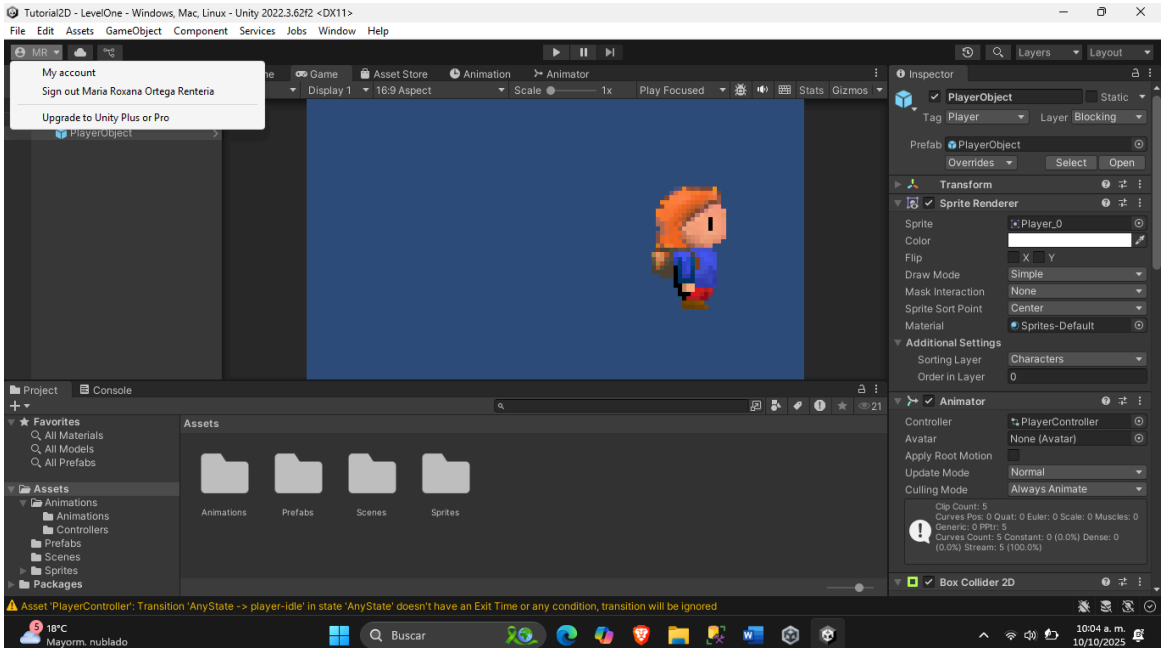
Capas

Las capas se utilizan para definir colecciones de **GameObjects**. Estas colecciones se utilizan en la detección de colisiones para determinar qué capas participan entre sí y así poder interactuar.

- Seleccionar en el menú desplegable Layers y seleccionar "Add layer". Debería ver la ventana Layers



Ahora seleccione el **PlayerObject** nuevamente para ver sus propiedades en el Inspector. Seleccione la capa de bloqueo que acabamos de crear en el menú desplegable.

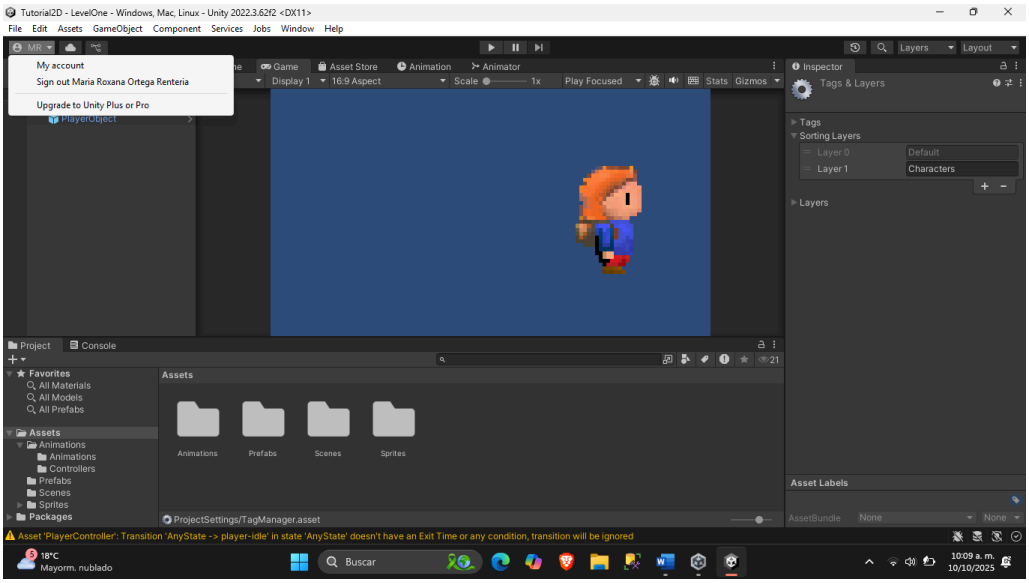


Más adelante, configuraremos nuestro juego para hacer cumplir la condición de que ciertos GameObjects no podrán atravesar ningún objeto en la capa de Bloqueo. Por ejemplo, el jugador estará en la capa de bloqueo, al igual que cualquier de las paredes, árboles o enemigos. Los enemigos no deben poder atravesar el jugador, y el jugador no debe poder atravesar paredes, árboles, o enemigos.

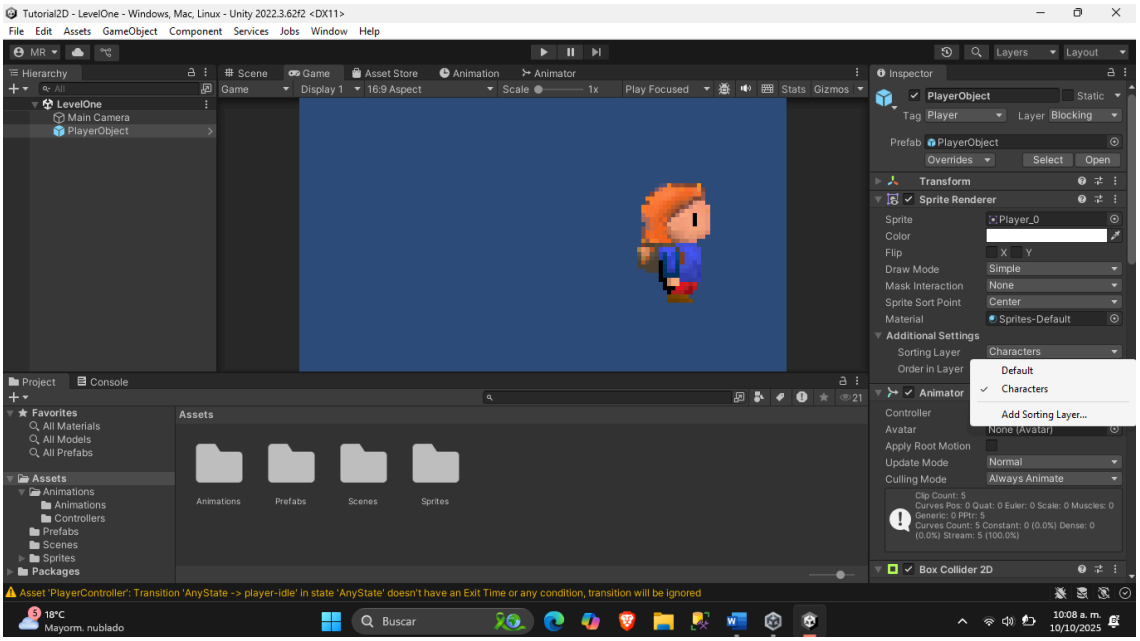
Sorting layers

Veamos ahora un tipo diferente de capa: Sorting Layers. Las Sorting Layers son diferentes a las capas regulares en que nos permiten decirle a la Unidad Motor en qué orden deben ser nuestros diversos Sprites 2D en la pantalla "Renderizado" o dibujado.

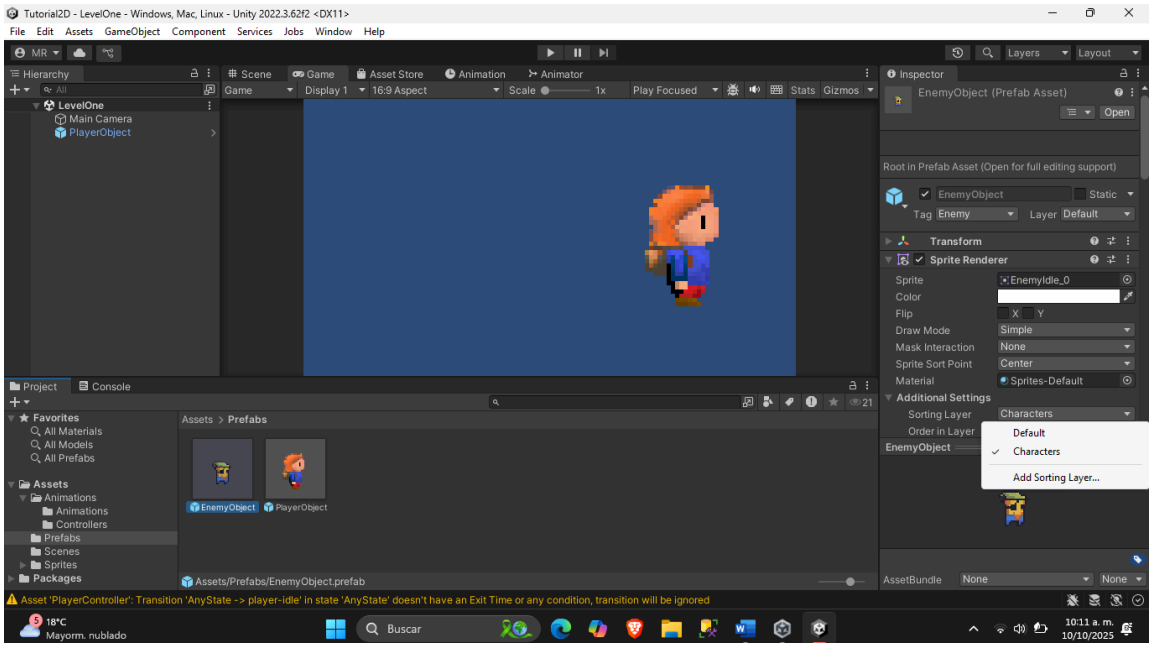
- Vamos a agregar una capa de ordenamiento llamada "**Caracteres**" que usaremos para nuestro jugador y todos los enemigos.
- En el componente Sprite Renderer en la ventana Inspector, seleccione la opción Sorting layer Menú desplegable de capa y seleccione "**Add Sorting Layer**".



- Agregue una capa de ordenamiento llamada "Characters", y luego haga clic en **PlayerObject** nuevamente para ver su Inspector y seleccione la nueva Capa de ordenamiento de caracteres del menú desplegable.



- Seleccione nuestro **EnemyObject** y establezca su Capa de ordenamiento en **Characters**, porque queremos que los enemigos también aparezcan en la parte superior de las cosas.

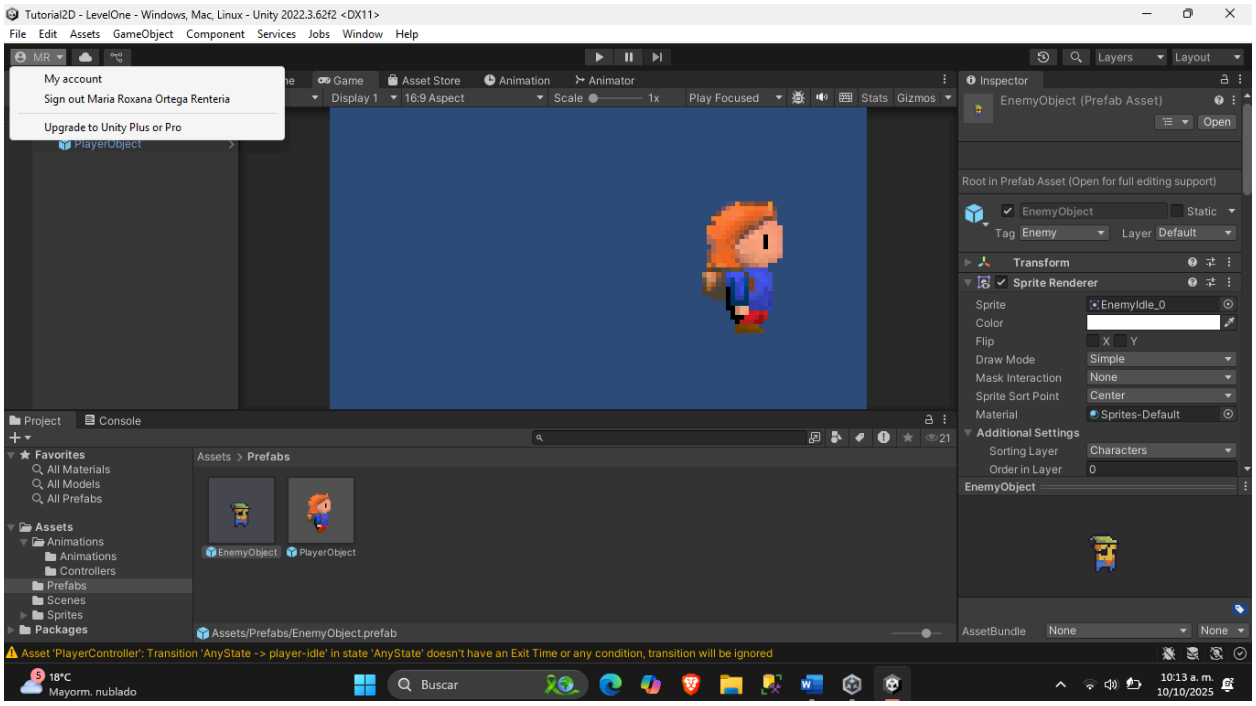


Prefabs

Unity te permite construir GameObjects con componentes integrados y luego crea algo llamado **"Prefab"** a partir de ese GameObject. Los prefabricados se pueden considerar como plantillas prefabricadas a partir de las cuales puede crear, o “instanciar” nuevas copias de GameObjects ya hechos. Este activo tiene una característica muy útil que le permite editar todos los prefabricados de una vez cambiando la plantilla prefabricada.

Usaremos esta técnica sencilla de prefabricados constantemente durante todo el proceso de construcción de nuestro juego.

- Primero, crea una carpeta llamada “**Prefabs**”en nuestra carpeta Assets en la vista Proyecto.
- Luego seleccione nuestro **PlayerObject** de la vista de Hierarchy y simplemente arrástralo a la carpeta **Prefabs**



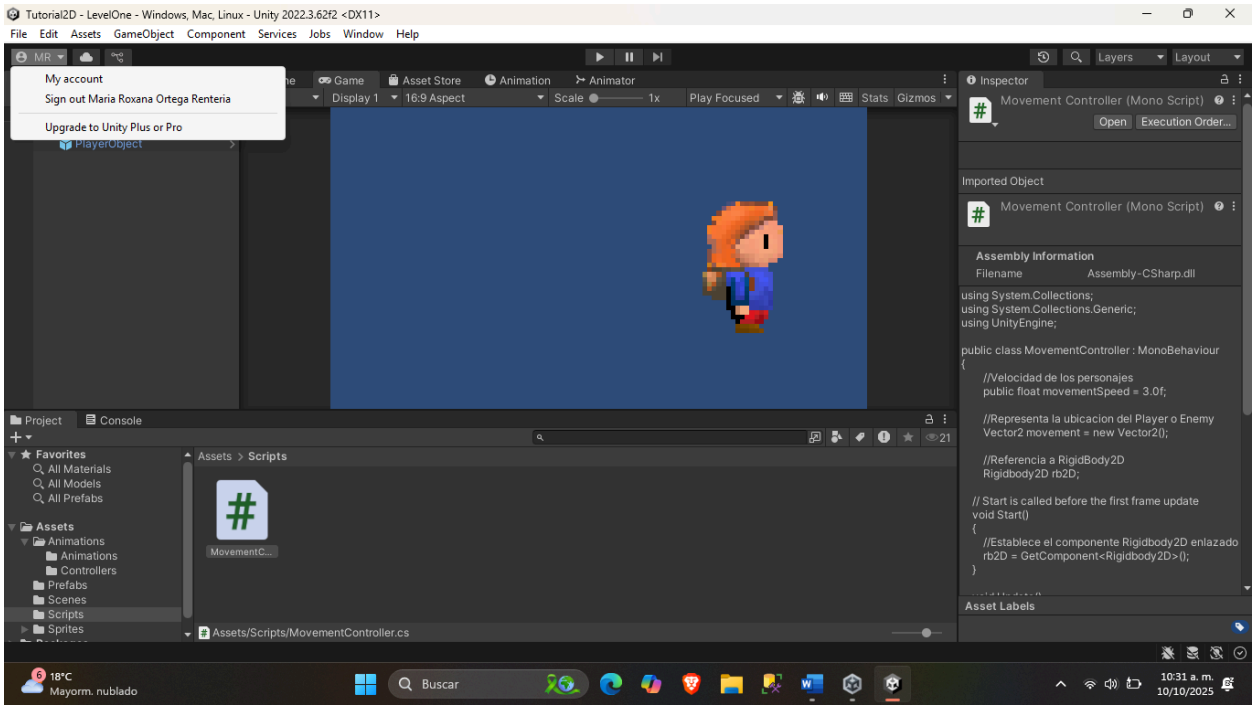
Ahora puedes eliminar de forma segura el PlayerObject de la vista de Hierarchy.

- Haga lo mismo con EnemyObject: arrástralo a la carpeta **Prefabs** y eliminar el EnemyObject original de la vista Hierarchy.

Scripts: Lógica para componentes

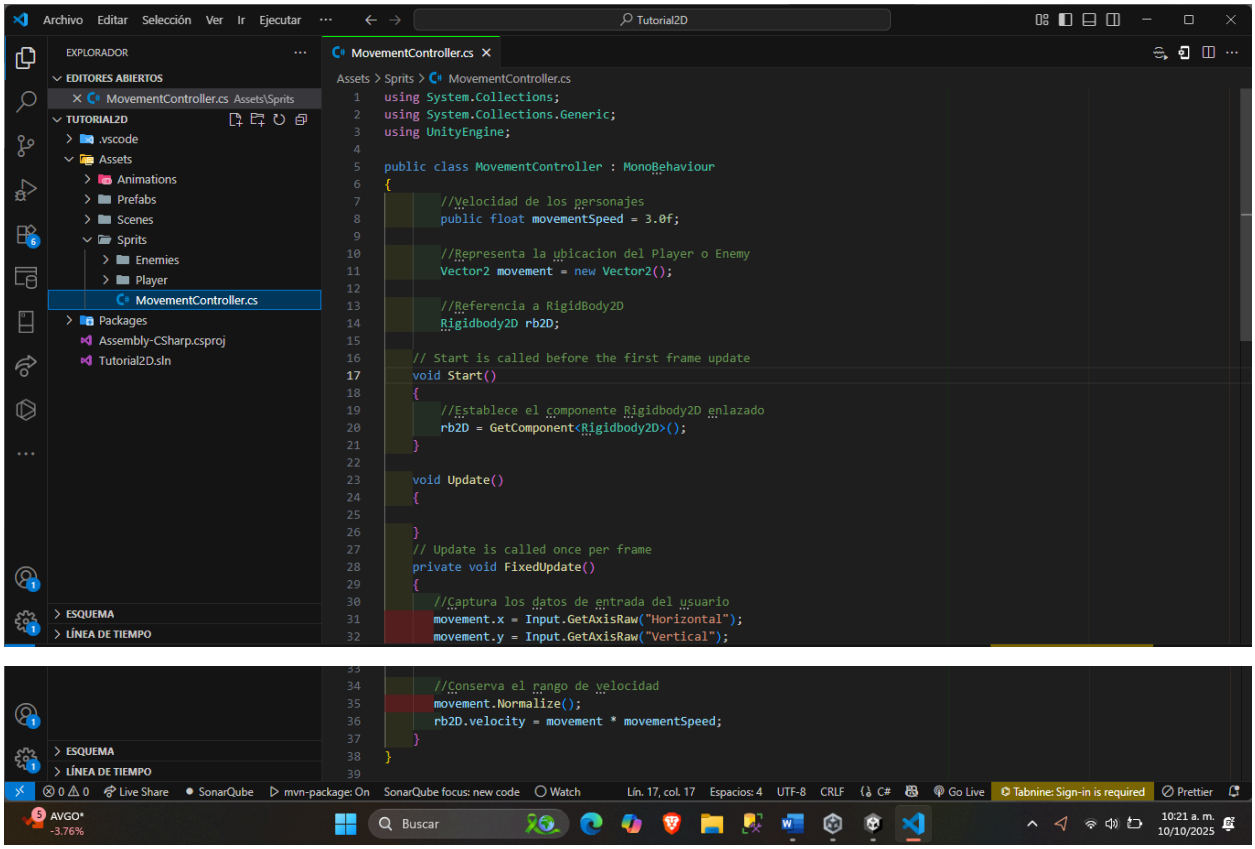
Entonces tenemos nuestro PlayerObject y tenemos nuestro EnemyObject. ¡Hagamos qué se mueven!

- Seleccione nuestro **PlayerObject** Prefab y arrástralo a la vista Hierarchy.
- Desplácese hasta la parte inferior del Inspector y presione el botón **Add Component**. Escriba la palabra Script y seleccione "New Script" y nombrarlo como "**MovementController**".
- Cree una nueva carpeta llamada "**Scripts**" en la vista Project. El nuevo script se habrá creado en la carpeta Assets de nivel superior en la vista Project. Arrastre el script MovementController a la carpeta Scripts y luego haga doble clic en él para abrirlo en Visual Studio.

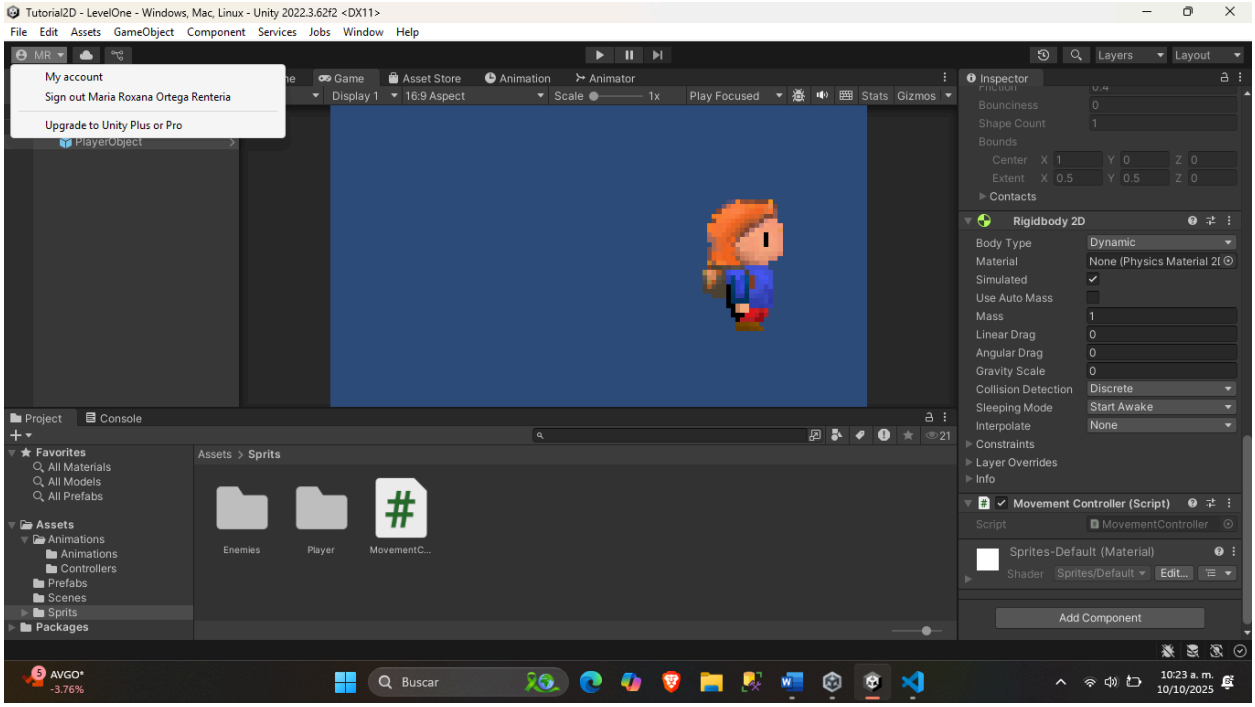


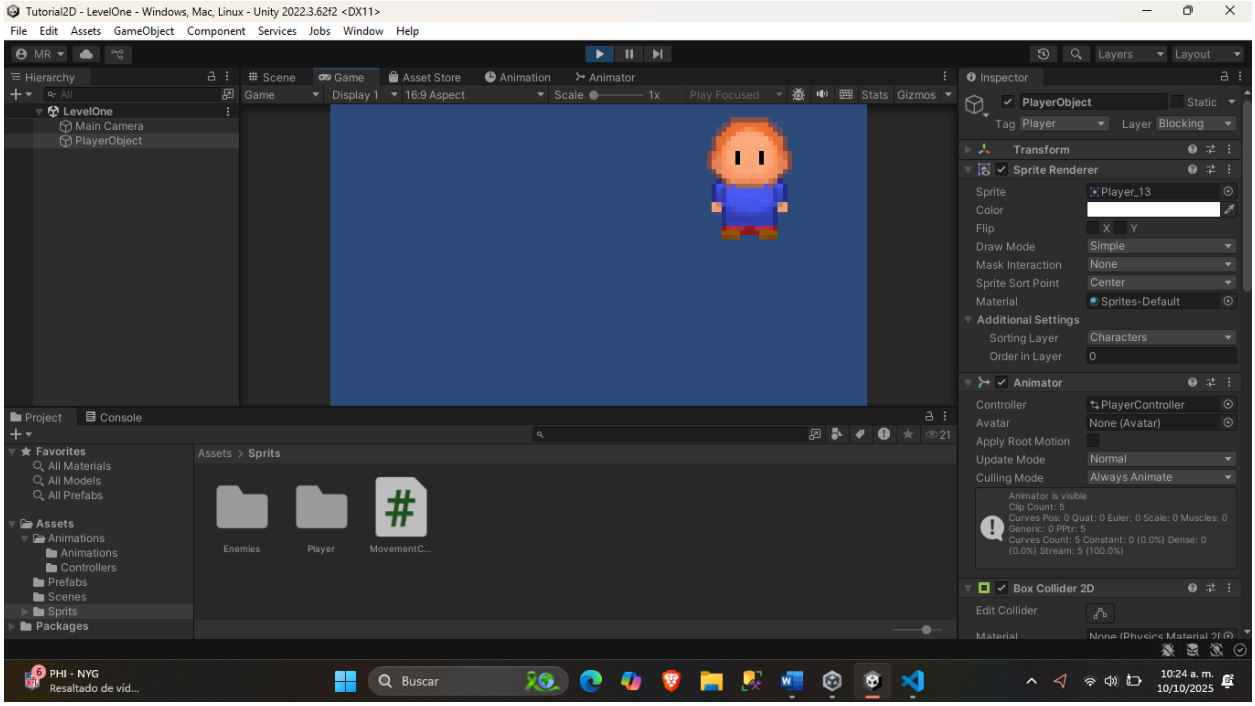
Es hora de programar nuestro primer Script. Los scripts en Unity están escritos en un lenguaje llamado C#. El espacio de nombres UnityEngine contiene muchas clases específicas de Unity como MonoBehaviour, GameObject, Rigidbody2D y BoxCollider2D.

- Agreguemos las siguientes propiedades a la clase



- Regrese al Editor de Unity y asegúrese de ver nuestro PlayerObject en la vista de Hierarchy. De lo contrario, arrastre PlayerObject desde la carpeta Prefabs en la vista de hierarchy.
- Para agregar el script a nuestro PlayerObject, arrastre el **MovementController** script de la carpeta Scripts, en PlayerObject en la jerarquía, o arrástralo al Inspector cuando el PlayerObject esté seleccionado.
- Ahora presione el botón Play. Deberías ver a nuestro personaje de jugador caminando en su lugar. Presione las teclas de flecha o W, A, S, D en su teclado y mírala moverse.


Tutorial2D - LevelOne - Windows, Mac, Linux - Unity 2022.3.62f2 <DX11>
File Edit Assets GameObject Component Services Jobs Window Help
My account
Sign out Maria Roxana Ortega Renteria
Upgrade to Unity Plus or Pro
Inspector
Rigidbody 2D
Movement Controller (Script)
Add Component
Project
Console
Assets > Sprits
Enemies
Player
Buscar


Tutorial2D - LevelOne - Windows, Mac, Linux - Unity 2022.3.62f2 <DX11>
File Edit Assets GameObject Component Services Jobs Window Help
Hierarchy
LevelOne
Main Camera
PlayerObject
Inspector
PlayerObject
Transform
Sprite Renderer
Animator
Box Collider 2D
Project
Console
Assets > Sprits
Enemies
Player
Buscar

Estados y Animaciones

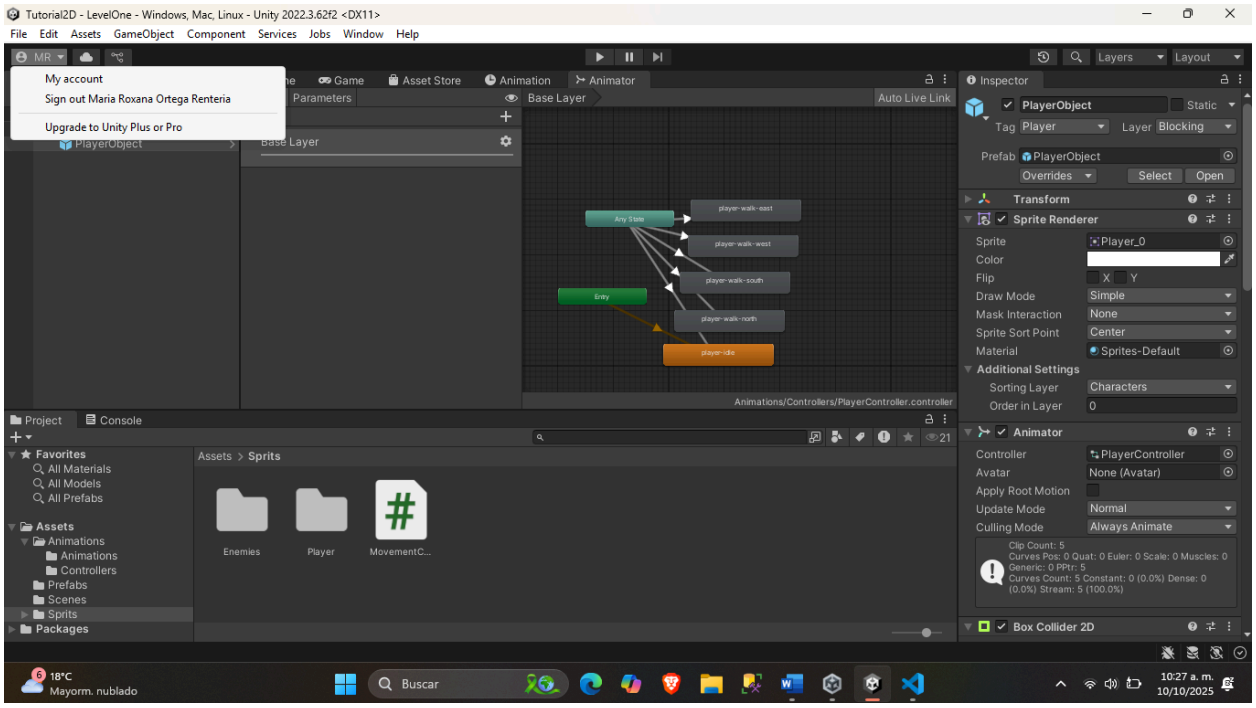
Más sobre máquina de estados

Ahora que sabemos cómo mover a nuestro personaje por la pantalla, vamos a hablar sobre cómo cambiar entre animaciones basadas en el estado actual del jugador.

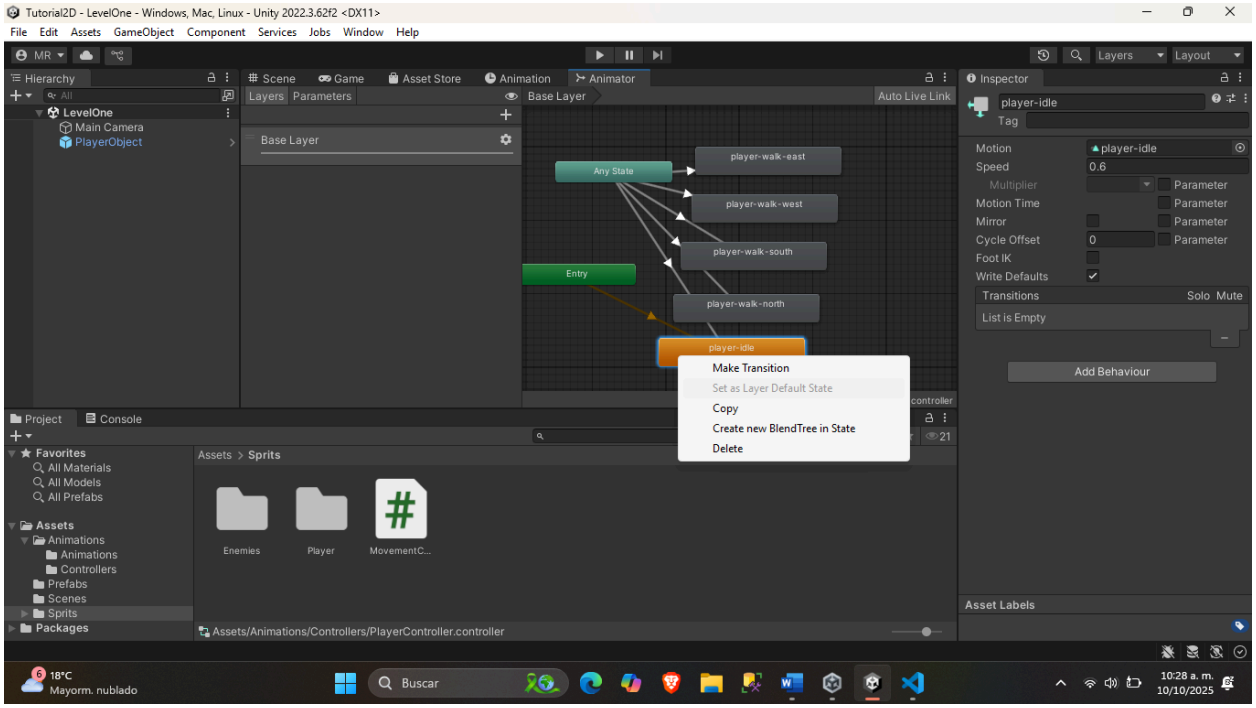
- Vaya a la carpeta Animations➤ Controllers y haga doble clic en el Objeto PlayerController.

Deberías estar mirando la ventana Animator, mostrando la máquina de estado que configuramos anteriormente. Como comentamos anteriormente, la máquina de estados de animación de Unity nos permite ver todos los estados de los jugadores y sus clips de animación asociados.

- Haga clic y arrastre sus objetos de estado de animación hasta que aparezcan en la pantalla, con el reproductor inactivo a un lado, y las animaciones de caminatas de jugador agrupadas

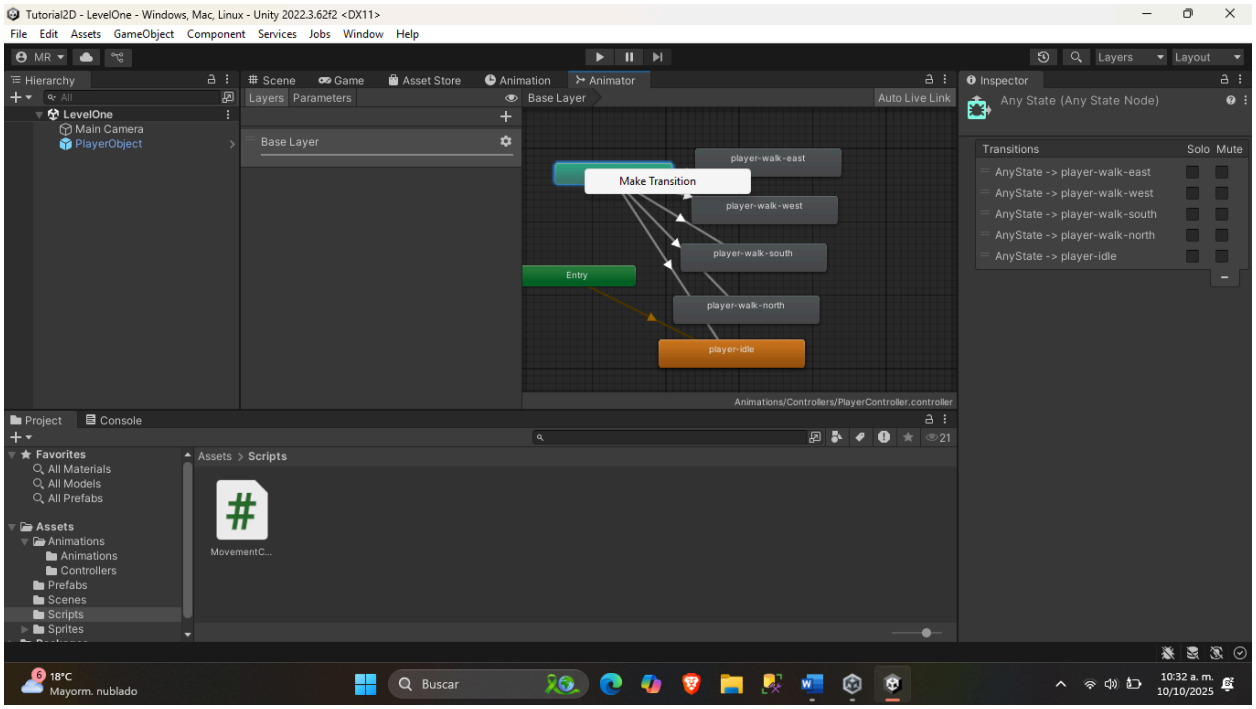


- El color naranja indica que es el estado predeterminado para este Animator. Seleccione y, a continuación, haga clic con el botón derecho en el estado de animación "player-idle" y seleccione "Set as Layer Default State".



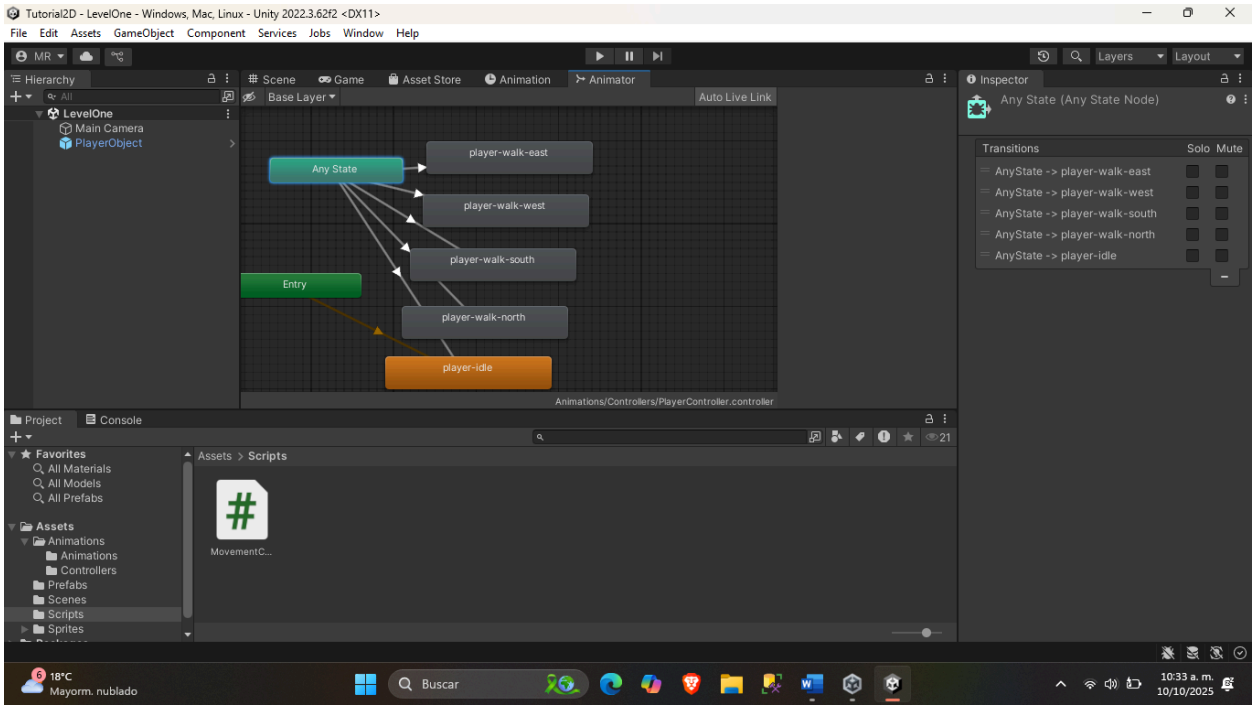
Queremos que player-idle sea el estado predeterminado porque cuando no estamos tocando una tecla direccional, queremos que el jugador mire al sur hacia el usuario en un estado inactivo.
Esto parecerá como si el personaje del jugador estuviera esperando al usuario.

- Ahora seleccione y haga clic con el botón derecho en "Any State" y seleccione "Make Transition" Aparecerá una línea con una flecha adjunta y siguiendo alrededor de su ratón. Haga click en "player-walk-east" para crear una transición entre el objeto Any State y player-walk-east.



- Ahora haga lo mismo para el resto de los estados de animación: haga clic con el botón derecho en Any State➢ Make Transition y seleccione cada uno de los Estados de animación para crear una transición.

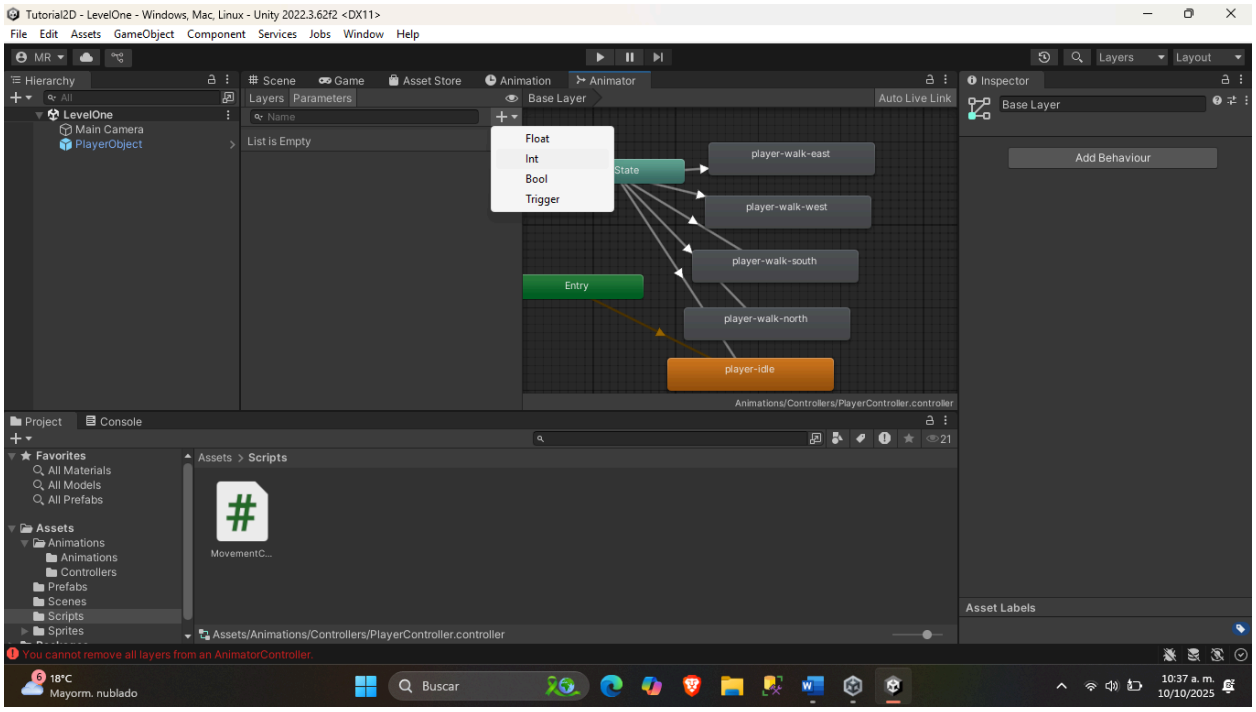
Debe crear un total de cinco flechas de transición blancas que apunten desde Any State a los cuatro estados de animación de caminata del jugador y al jugador inactivo player-idle. También debería haber un estado predeterminado de color naranja flecha desde el estado de animación de entrada, que conduce a la animación de jugador inactivo Estado.

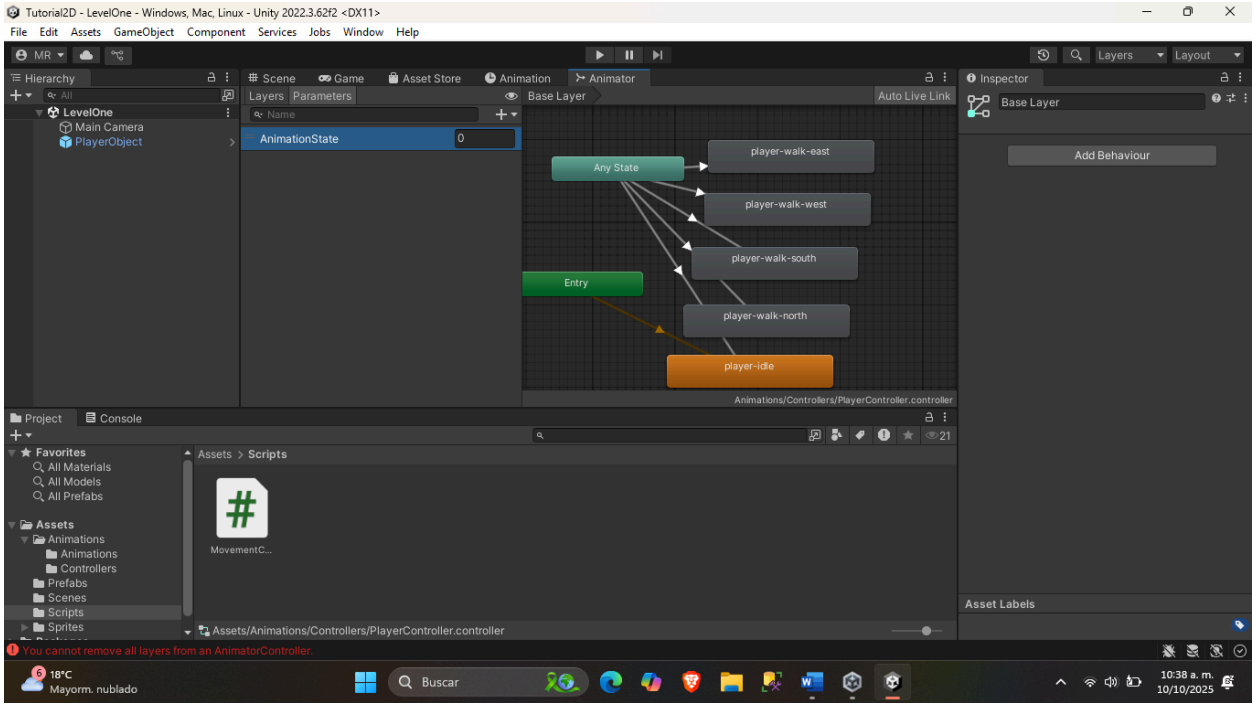


Parámetros de estado de Animación

Los parámetros de animación son variables definidas en el Controlador de animación y son utilizados por scripts para controlar la animación. Vamos a utilizar este parámetro de animación que creamos en nuestras Transiciones y en nuestro script **MovementController** para controlar el PlayerObject y hacer que camine por la pantalla.

- Seleccione la pestaña Parámetros en el lado izquierdo de la Ventana Animator. Presione el símbolo **+** y seleccione "Int" en la lista desplegable. Cambie el nombre del parámetro de animación creado a “AnimationState”.

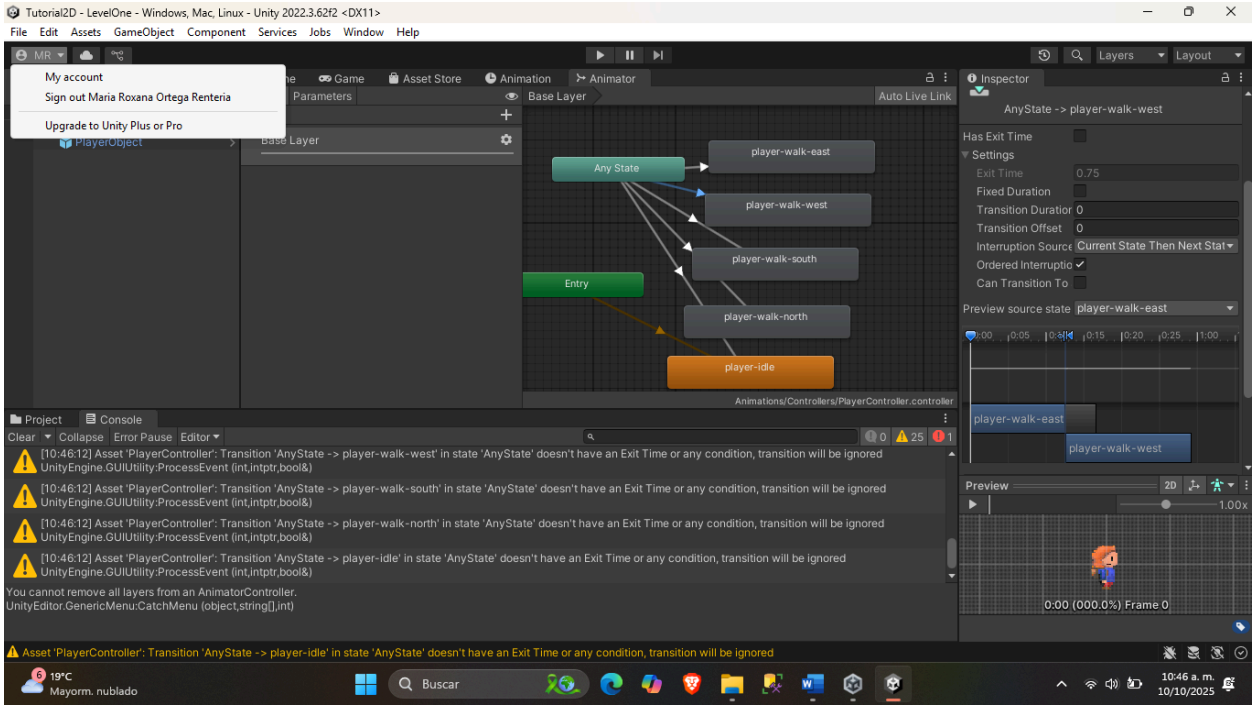

Tutorial2D - LevelOne - Windows, Mac, Linux - Unity 2022.3.62f2 <DX11>
File Edit Assets GameObject Component Services Jobs Window Help
Hierarchy
LevelOne
Main Camera
PlayerObject
Scene Game Asset Store Animation Animator
Layers Parameters
Base Layer
Auto Live Link
List is Empty
Float
Int
Bool
Trigger
State
player-walk-east
player-walk-west
player-walk-south
player-walk-north
player-idle
Entry
Animations/Controllers/PlayerController.controller
Inspector
Base Layer
Add Behaviour
Asset Labels
Project Console
Favorites
All Materials
All Models
All Prefabs
Assets
Animations
Controllers
Prefabs
Scenes
Scripts
Sprites
Assets > Scripts
MovementC...
Assets/Animations/Controllers/PlayerController.controller
You cannot remove all layers from an AnimatorController.
18°C
Mayorm. nublado
Buscar
10:37 a. m.
10/10/2025


Tutorial2D - LevelOne - Windows, Mac, Linux - Unity 2022.3.62f2 <DX11>
File Edit Assets GameObject Component Services Jobs Window Help
Hierarchy
LevelOne
Main Camera
PlayerObject
Scene Game Asset Store Animation Animator
Layers Parameters
Base Layer
Auto Live Link
AnimationState
0
Any State
player-walk-east
player-walk-west
player-walk-south
player-walk-north
player-idle
Entry
Animations/Controllers/PlayerController.controller
Inspector
Base Layer
Add Behaviour
Asset Labels
Project Console
Favorites
All Materials
All Models
All Prefabs
Assets
Animations
Controllers
Prefabs
Scenes
Scripts
Sprites
Assets > Scripts
MovementC...
Assets/Animations/Controllers/PlayerController.controller
You cannot remove all layers from an AnimatorController.
18°C
Mayorm. nublado
Buscar
10:38 a. m.
10/10/2025

Vamos a establecer el parámetro de animación en cada transición a una condición específica. Si durante el juego esta condición es cierta, entonces el animador pasará a ese estado de animación y al correspondiente se reproducirá el clip de animación. Debido a que este componente Animator se adjunta a **PlayerObject**, los clips de animación se mostrarán en el componente Transform la ubicación del componente en la escena. Usamos un script para configurar esta animación. La condición del parámetro debe ser verdadera y desencadenar la transición de estado.

- Seleccione la línea de transición blanca que conecta cualquier estado con el estado en la Inspector, cambie la configuración para que coincida.

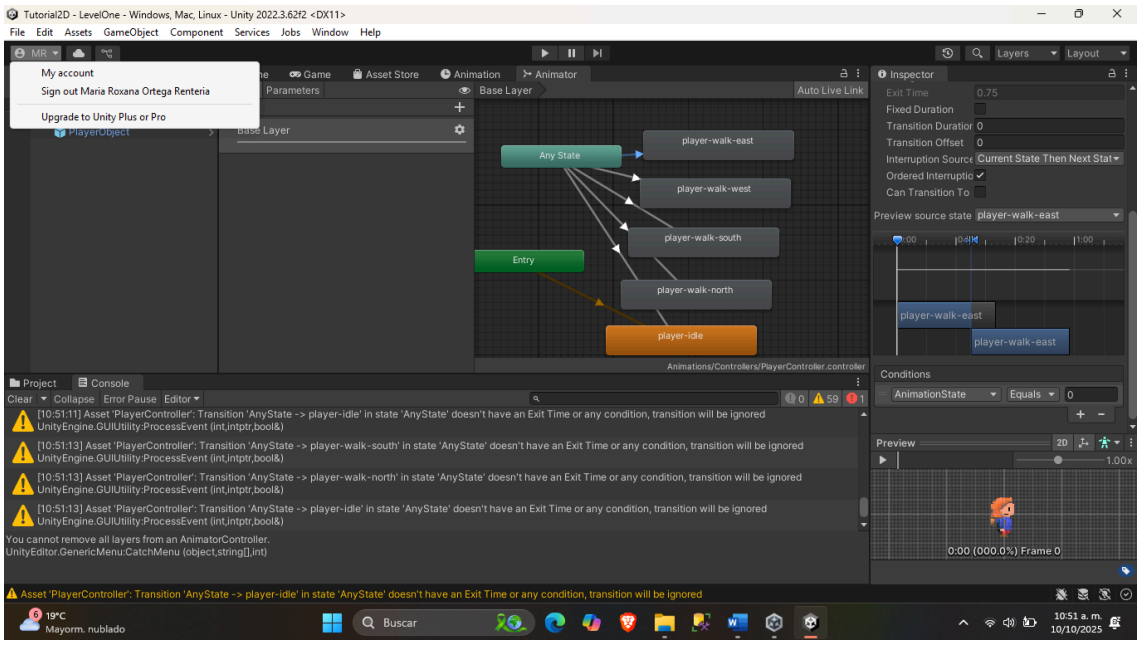
  Queremos desmarcar casillas como Exit Time, Fixed duration y Can Transition to Self. Asegúrese de establecer la Transition duration (%) en 0 y fuente de interrupción a "Current State Then Next State".

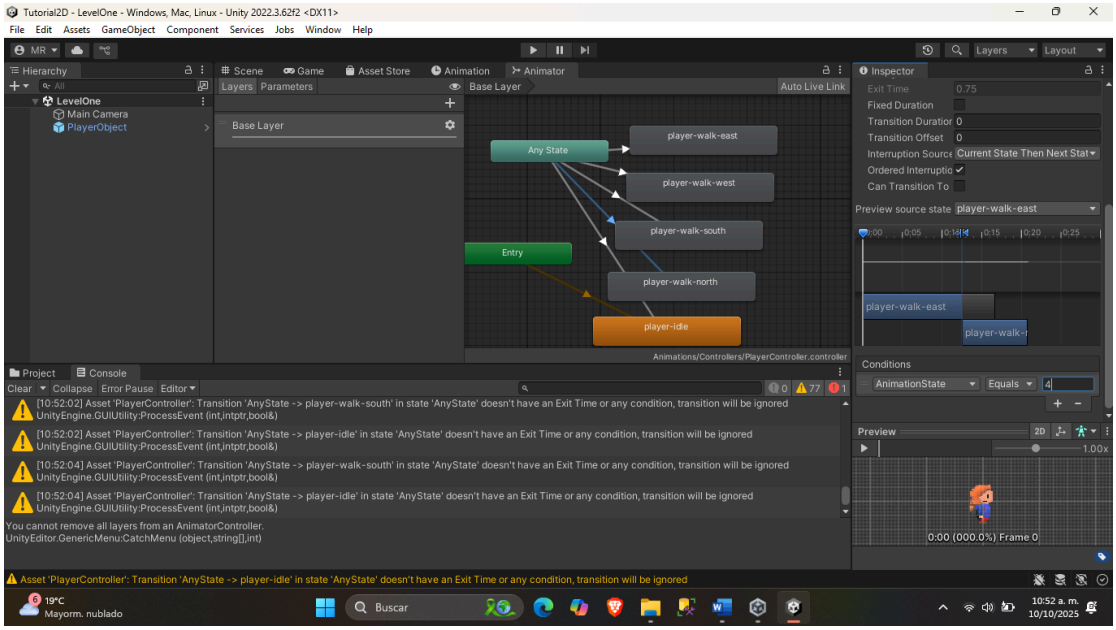


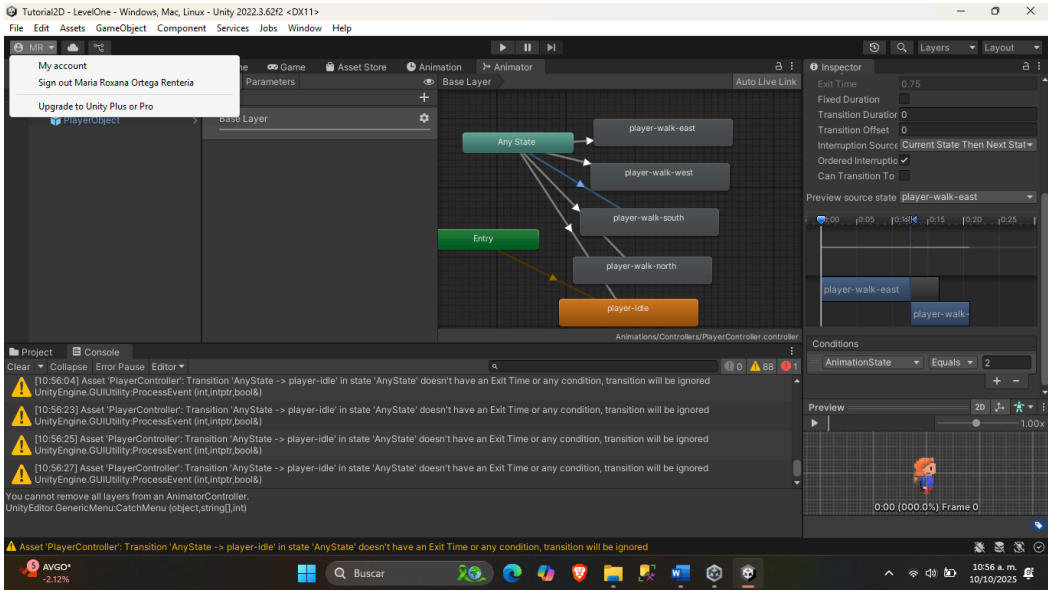
Desmarque Has Exit Time porque queremos interrumpir una animación si nuestro usuario presiona una tecla diferente. Si dejamos marcado Has Exit Time, entonces la animación tendría que terminar de reproducirse hasta que el % ingresado en la salida Exit time, antes de que pueda comenzar la siguiente, y eso daría como resultado una mala experiencia del jugador.

- En la parte inferior del inspector, verá un área titulada "Conditions". Haga clic en el símbolo + en la parte inferior derecha y seleccione AnimationState ➢ Equals, e ingrese 1

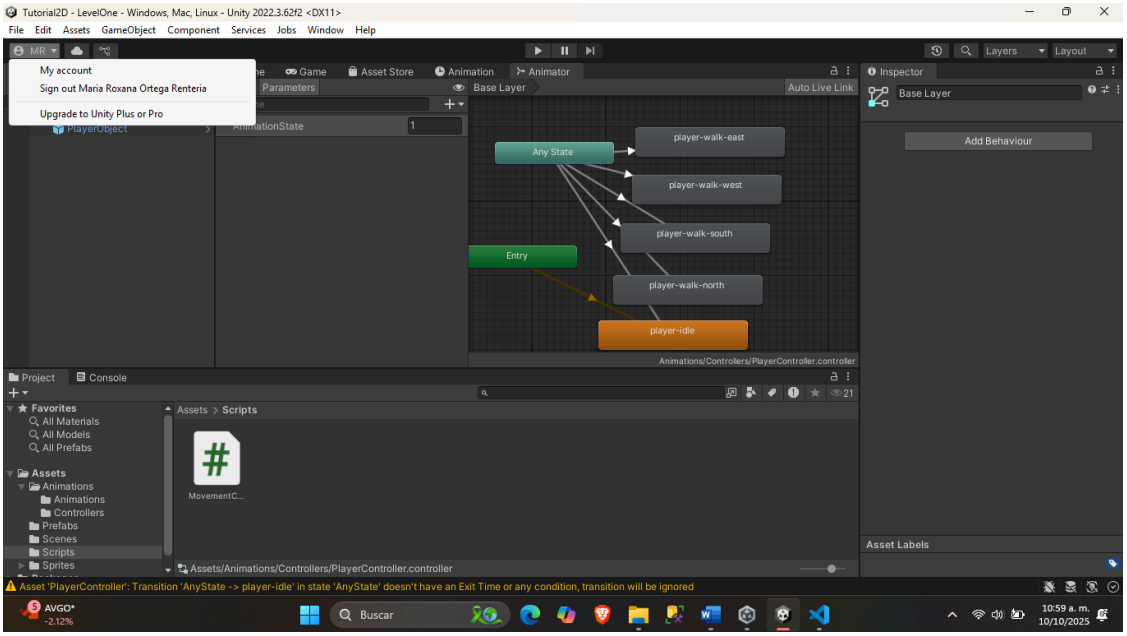
Transición	Condición
Any State to **player-walk-east**	**AnimationState** Equals = **1**
Any State to **player-walk-west**	**AnimationState** Equals = **3**
Any State to **player-walk-north**	**AnimationState** Equals = **4**
Any State to **player-walk-south**	**AnimationState** Equals = **2**
Any State to **player-idle**	**AnimationState** Equals = **0**


Tutorial2D - LevelOne - Windows, Mac, Linux - Unity 2022.3.62f2 <DX11>
File Edit Assets GameObject Component Services Jobs Window Help
My account
Sign out Maria Roxana Ortega Renteria
Upgrade to Unity Plus or Pro
Game
Asset Store
Animation
Animator
Parameters
Base Layer
Auto Live Link
Any State
player-walk-east
player-walk-west
player-walk-south
Entry
player-walk-north
player-idle
Animations/Controllers/PlayerController.controller
Inspector
Exit Time 0.75
Fixed Duration
Transition Duration 0
Transition Offset 0
Interruption Source Current State Then Next Stat
Ordered Interruptio
Can Transition To
Preview source state player-walk-east
Conditions
AnimationState Equals 0
Preview
1.00x
0:00 (000.0%) Frame 0
Project
Console
Clear Collapse Error Pause Editor
[10:51:11] Asset 'PlayerController': Transition 'AnyState -> player-idle' in state 'AnyState' doesn't have an Exit Time or any condition, transition will be ignored
UnityEngine.GUIUtility:ProcessEvent (int,intptr,bool&)
[10:51:13] Asset 'PlayerController': Transition 'AnyState -> player-walk-south' in state 'AnyState' doesn't have an Exit Time or any condition, transition will be ignored
UnityEngine.GUIUtility:ProcessEvent (int,intptr,bool&)
[10:51:13] Asset 'PlayerController': Transition 'AnyState -> player-walk-north' in state 'AnyState' doesn't have an Exit Time or any condition, transition will be ignored
UnityEngine.GUIUtility:ProcessEvent (int,intptr,bool&)
[10:51:13] Asset 'PlayerController': Transition 'AnyState -> player-idle' in state 'AnyState' doesn't have an Exit Time or any condition, transition will be ignored
UnityEngine.GUIUtility:ProcessEvent (int,intptr,bool&)
You cannot remove all layers from an AnimatorController.
UnityEditor.GenericMenu:CatchMenu (object,string[],int)
Asset 'PlayerController': Transition 'AnyState -> player-idle' in state 'AnyState' doesn't have an Exit Time or any condition, transition will be ignored
19°C
Mayorm. nublado
Buscar
10:51 a. m.
10/10/2025

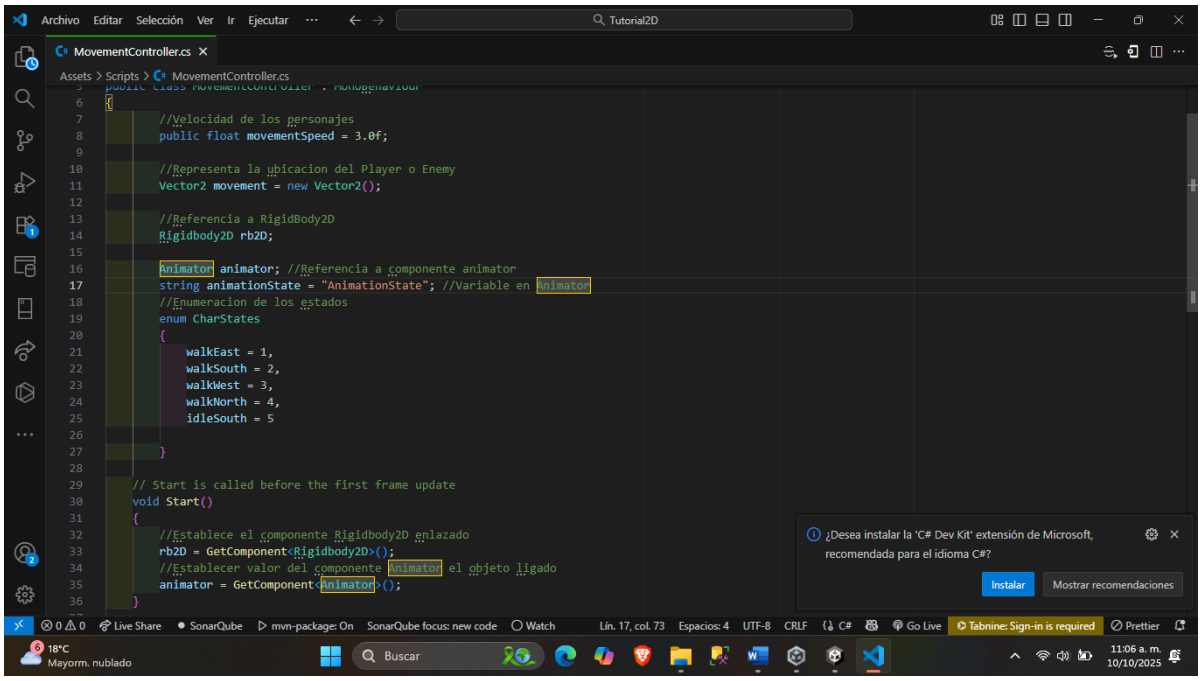

Tutorial2D - LevelOne - Windows, Mac, Linux - Unity 2022.3.62f2 <DX11>
File Edit Assets GameObject Component Services Jobs Window Help
Hierarchy
LevelOne
Main Camera
PlayerObject
Scene Game Asset Store Animation Animator
Layers Parameters
Base Layer
Auto Live Link
Any State
player-walk-east
player-walk-west
player-walk-south
Entry
player-walk-north
player-idle
Animations/Controllers/PlayerController.controller
Inspector
Exit Time 0.75
Fixed Duration
Transition Duration 0
Transition Offset 0
Interruption Source Current State Then Next Stat
Ordered Interruptio
Can Transition To
Preview source state player-walk-east
Conditions
AnimationState Equals 4
Preview
2D
1.00x
0:00 (000.0%) Frame 0
Project Console
Clear Collapse Error Pause Editor
[10:52:02] Asset 'PlayerController': Transition 'AnyState -> player-walk-south' in state 'AnyState' doesn't have an Exit Time or any condition, transition will be ignored
UnityEngine.GUIUtility:ProcessEvent (int,intptr,bool&)
[10:52:02] Asset 'PlayerController': Transition 'AnyState -> player-idle' in state 'AnyState' doesn't have an Exit Time or any condition, transition will be ignored
UnityEngine.GUIUtility:ProcessEvent (int,intptr,bool&)
[10:52:04] Asset 'PlayerController': Transition 'AnyState -> player-walk-south' in state 'AnyState' doesn't have an Exit Time or any condition, transition will be ignored
UnityEngine.GUIUtility:ProcessEvent (int,intptr,bool&)
[10:52:04] Asset 'PlayerController': Transition 'AnyState -> player-idle' in state 'AnyState' doesn't have an Exit Time or any condition, transition will be ignored
UnityEngine.GUIUtility:ProcessEvent (int,intptr,bool&)
You cannot remove all layers from an AnimatorController.
UnityEditor.GenericMenu:CatchMenu (object,string[],int)
Asset 'PlayerController': Transition 'AnyState -> player-idle' in state 'AnyState' doesn't have an Exit Time or any condition, transition will be ignored
19°C
Mayorm. nublado
Buscar
10:52 a. m.
10/10/2025



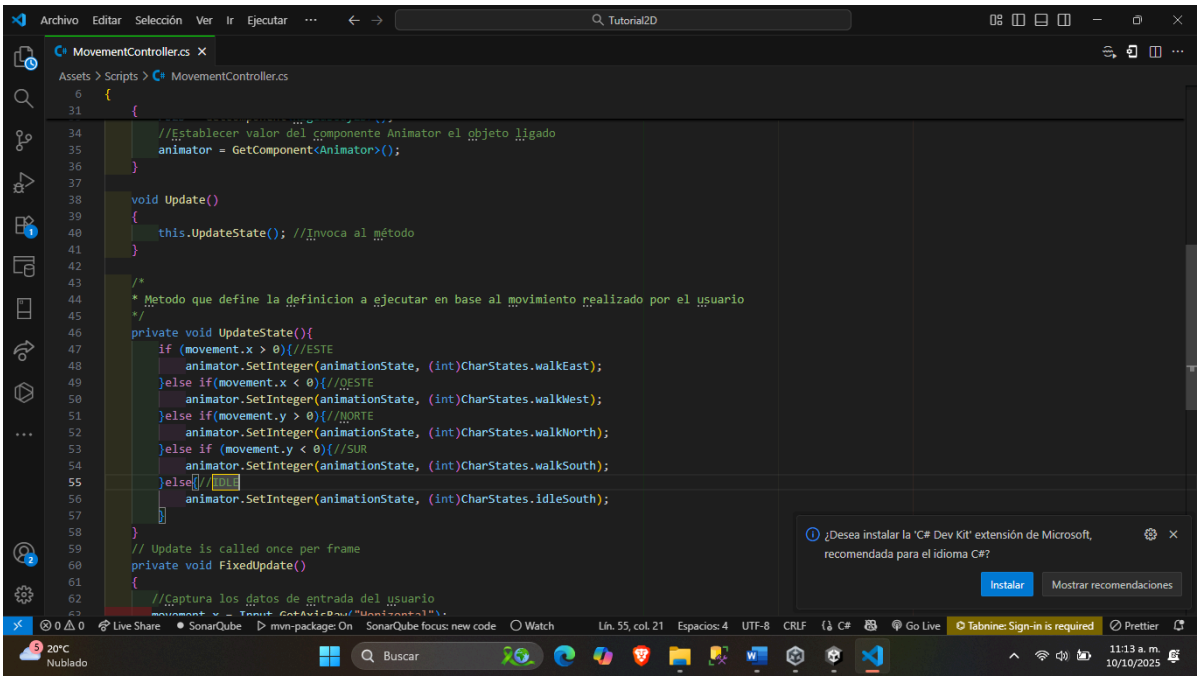
- Lo siguiente que vamos a hacer es establecer que el parámetro **AnimationState** igual a 1 en nuestro script. Regrese a Visual Studio y abra nuestro Script MovementController.cs.



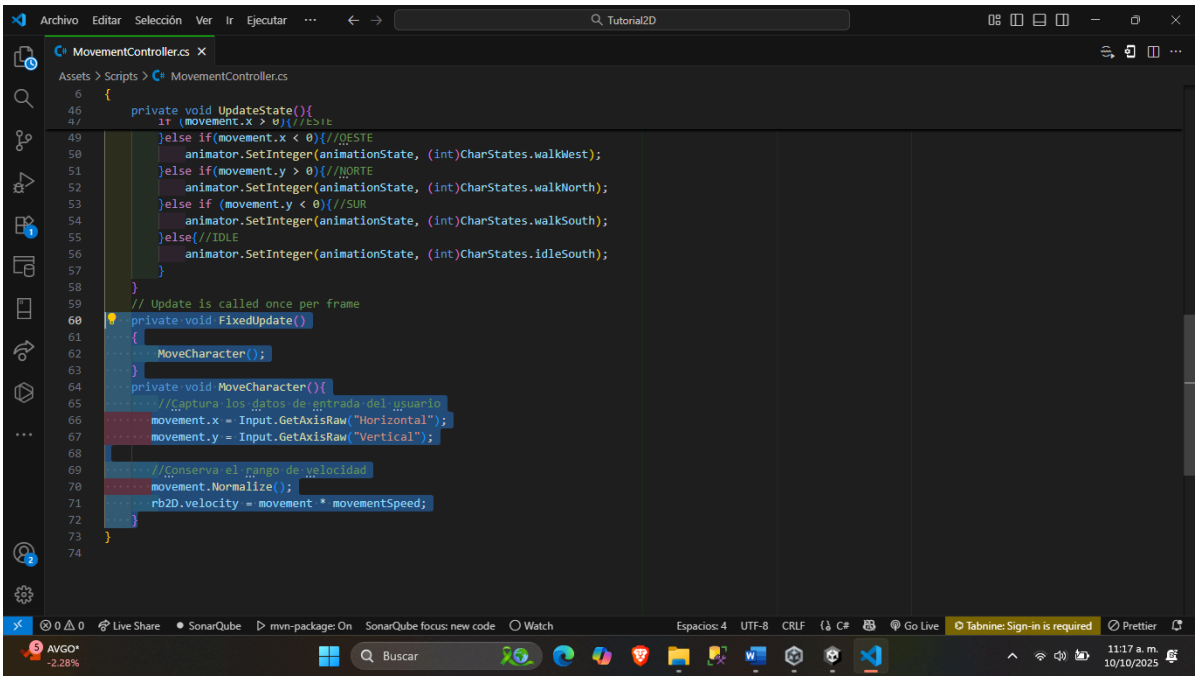
- Agreguemos la referencia del objeto Animator; referencia de la variable qué guarda los estados y la definición de los estados



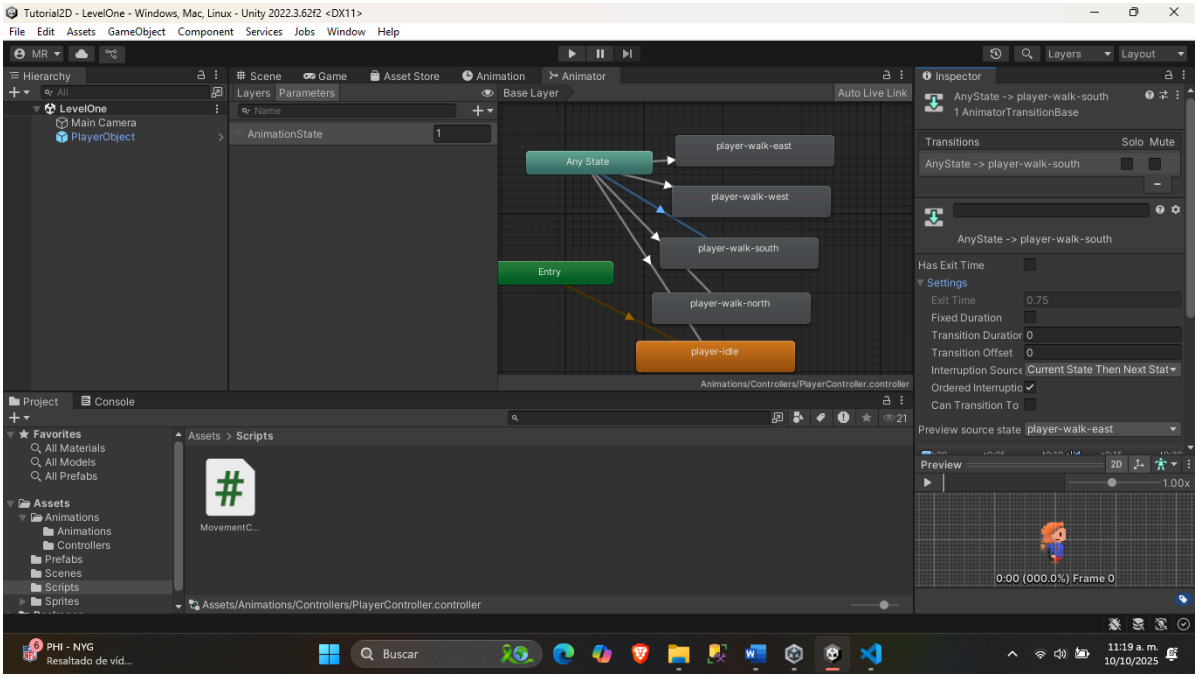
- Inicializamos en el método start el valor del componente Animator del objeto enlazado Modifiquemos el método Update donde se invoca a otro método qué define en base a las entradas WASD por el usuario


Archivo Editar Selección Ver Ir Ejecutar
Tutorial2D
MovementController.cs
Assets > Scripts > MovementController.cs
//Establecer valor del componente Animator el objeto ligado
animator = GetComponent<Animator>();
void Update()
this.UpdateState(); //Invoca al método
/*
* Metodo que define la definicion a ejecutar en base al movimiento realizado por el usuario
*/
private void UpdateState(){
if (movement.x > 0){//ESTE
animator.SetInteger(animationState, (int)CharStates.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);
}
// Update is called once per frame
private void FixedUpdate()
//Captura los datos de entrada del usuario
¿Desea instalar la 'C# Dev Kit' extensión de Microsoft, recomendada para el idioma C#?
Instalar
Mostrar recomendaciones
Live Share
SonarQube
mvn-package: On
SonarQube focus: new code
Watch
Lín. 55, col. 21
Espacios: 4
UTF-8
CRLF
C#
Go Live
Tabnine: Sign-in is required
Prettier
20°C
Nublado
Buscar
11:13 a. m.
10/10/2025

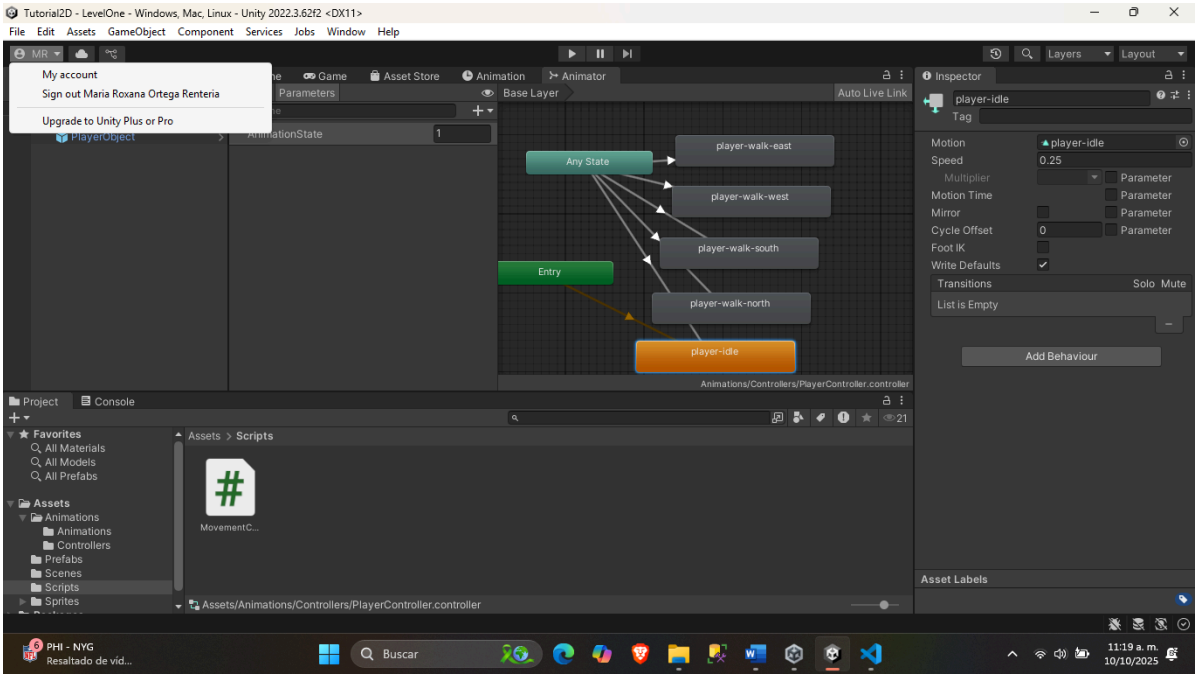
Modifiquemos el método FixedUpdate



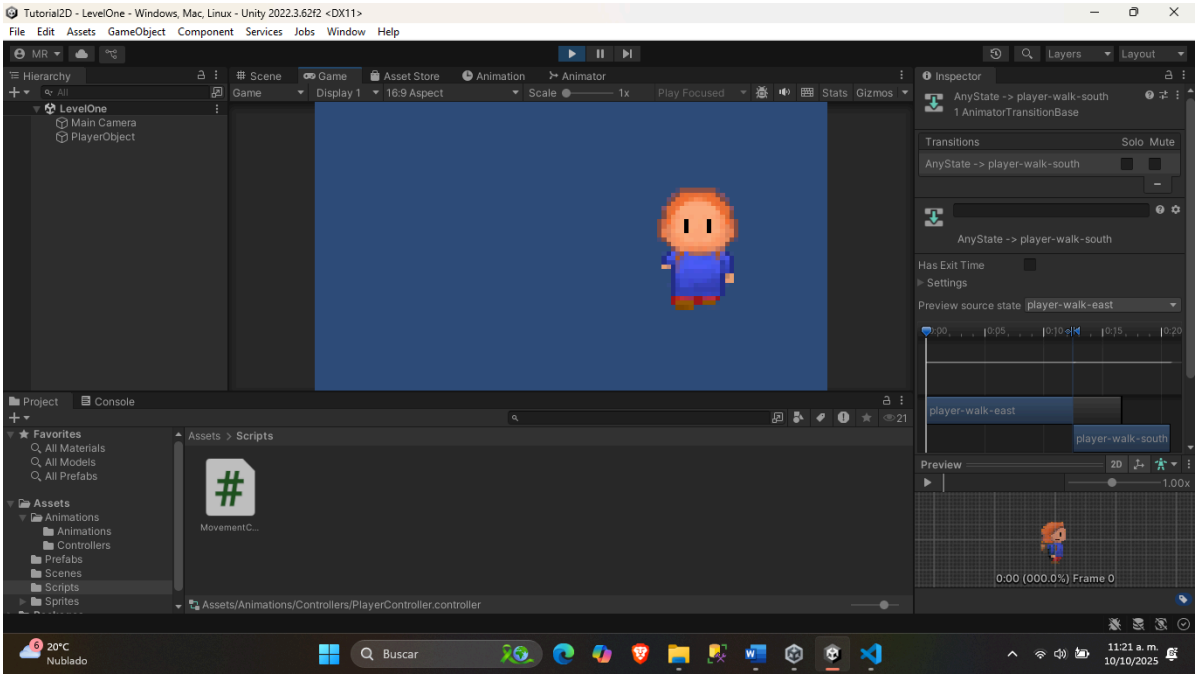
- Regresa al Animator de Unity y verifica cada estado de transición. A medida que avanza por cada flecha de transición, recuerde desmarcar cuadros como Exit Time, Fixed Duration, Can Transition to Self y establecer Transition Duration (%) a 0.



Seleccione cada estado de animación de caminata del jugador objeto y ajuste la velocidad a 0,6, y ajuste player-idle a 0.25. Esto hará que las animaciones de nuestros jugadores se vean bien.



A. Ahora ha configurado una gran parte de las animaciones del reproductor necesarias para nuestro juego. Pulsamos el botón Play y movemos nuestro personaje por la pantalla con las teclas de flecha o W, A, S, D.


Tutorial2D - LevelOne - Windows, Mac, Linux - Unity 2022.3.62f2 <DX11>
File Edit Assets GameObject Component Services Jobs Window Help
Hierarchy
LevelOne
Main Camera
PlayerObject
Scene Game Asset Store Animation Animator
Inspector
AnyState -> player-walk-south
1 AnimatorTransitionBase
Transitions Solo Mute
AnyState -> player-walk-south
Has Exit Time
Settings
Preview source state player-walk-east
player-walk-east
player-walk-south
Preview
0:00 (000.0%) Frame 0
Project Console
Favorites
All Materials
All Models
All Prefabs
Assets
Animations
Controllers
Prefabs
Scenes
Scripts
Sprites
Assets > Scripts
MovementC...
Assets/Animations/Controllers/PlayerController.controller
20°C
Nublado
Buscar
11:21 a. m.
10/10/2025