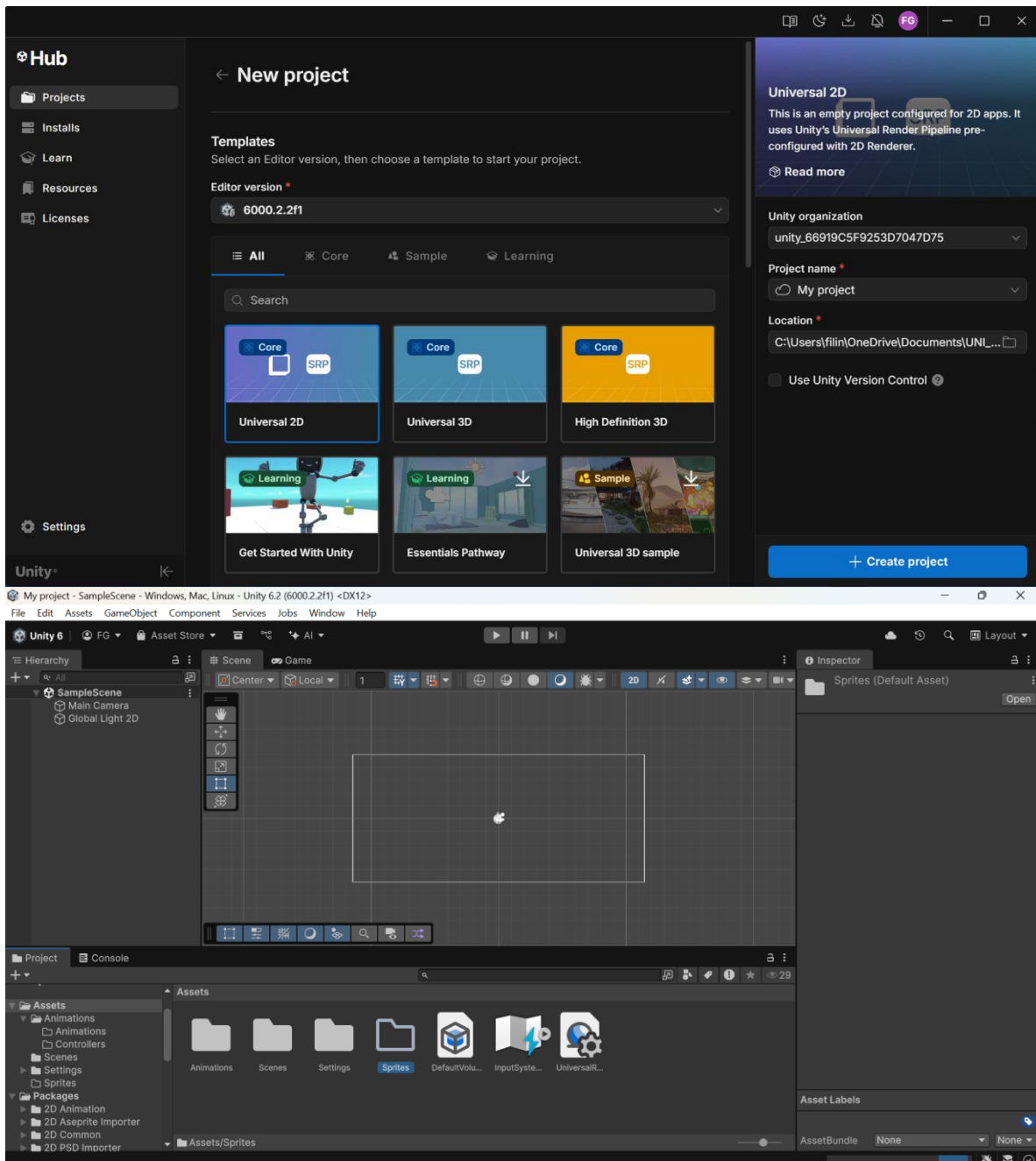
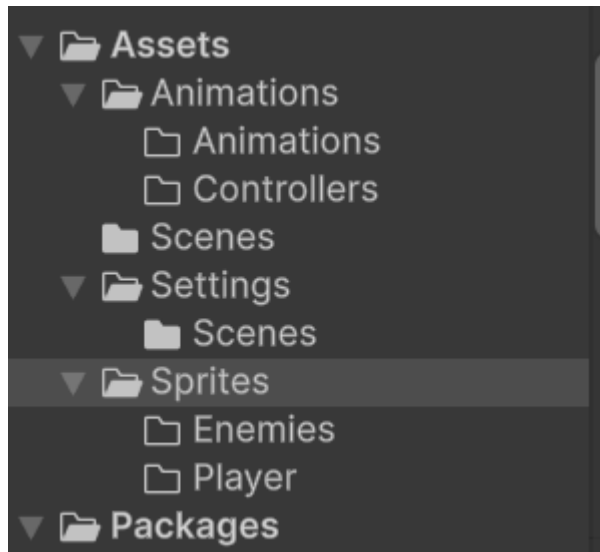


## Crear el proyecto



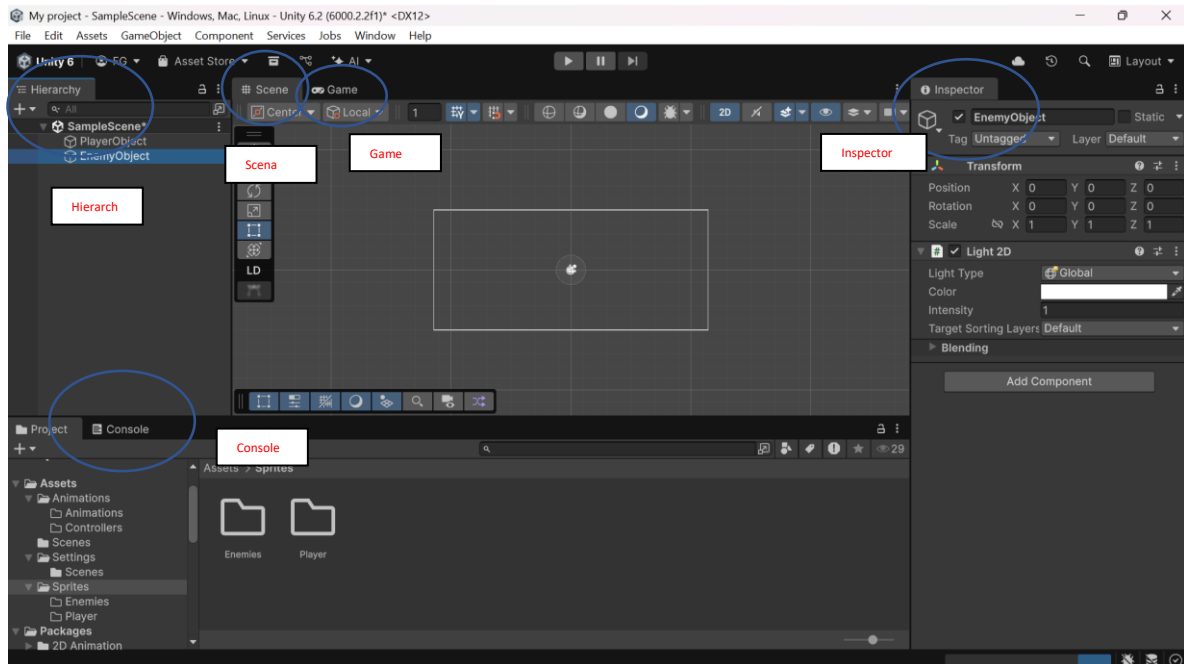
Se organizan las carpetas



## Cuadro de Vistas y Funciones de Unity

Vista	Función Principal	¿Qué puedes hacer ahí?
Scene	Editar y diseñar el entorno del juego.	Colocar, mover, rotar y escalar objetos (GameObjects) en el mundo. Es tu lienzo para construir los niveles y escenas.
Game	Previsualizar y probar cómo se verá y comportará el juego desde la perspectiva del jugador.	Ver la escena tal como la vería la cámara principal (o el jugador). Es esencial para hacer pruebas (playtesting) y ajustar la interfaz de usuario (UI).
Hierarchy	Gestionar la lista de todos los objetos (GameObjects) presentes en la escena actual.	Seleccionar, organizar, crear y eliminar GameObjects. También permite crear estructuras padre-hijo arrastrando objetos sobre otros para agruparlos.
Inspector	Ver y editar las propiedades y componentes de cualquier GameObject seleccionado.	Modificar transformaciones (posición, rotación, escala), añadir componentes (como scripts, físicas, colliders), y ajustar sus configuraciones y parámetros.
Project	Gestionar todos los archivos y recursos (assets) del proyecto.	Crear carpetas, importar assets (modelos, texturas, sonidos, scripts), organizarlos y seleccionarlos para usarlos en la escena. Es el explorador de archivos de tu proyecto.
Console	Visualizar mensajes, advertencias (warnings) y errores generados por Unity o por tus scripts.	Depurar tu código, ver los resultados de <code>Debug.Log()</code> , identificar errores de compilación y limpiar mensajes para mantener un flujo de trabajo ordenado.

Seleccionar cada parte de unity



1. ¿Modificar propiedades?

Inspector

2. ¿Ver el juego?

Vista Game

3. ¿Ver errores?

Console

4. Diferencia Scene vs. Game:

Scene: Es el taller para editar.

Game: Es el simulador para probar lo que ve el jugador.

5. Create Empty:

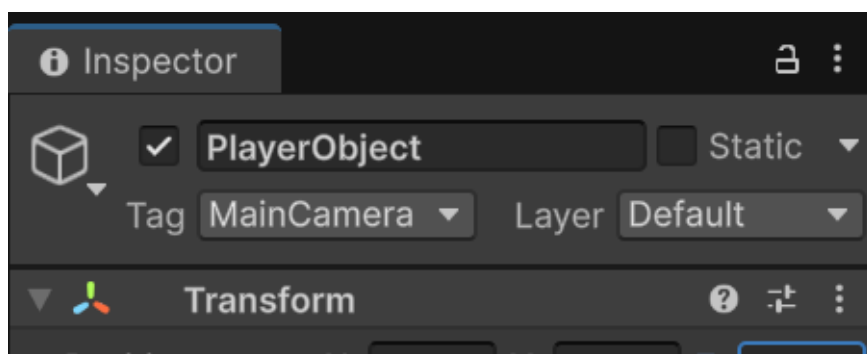
Es un GameObject vacío e invisible. Solo tiene posición, rotación y escala. Se usa para:

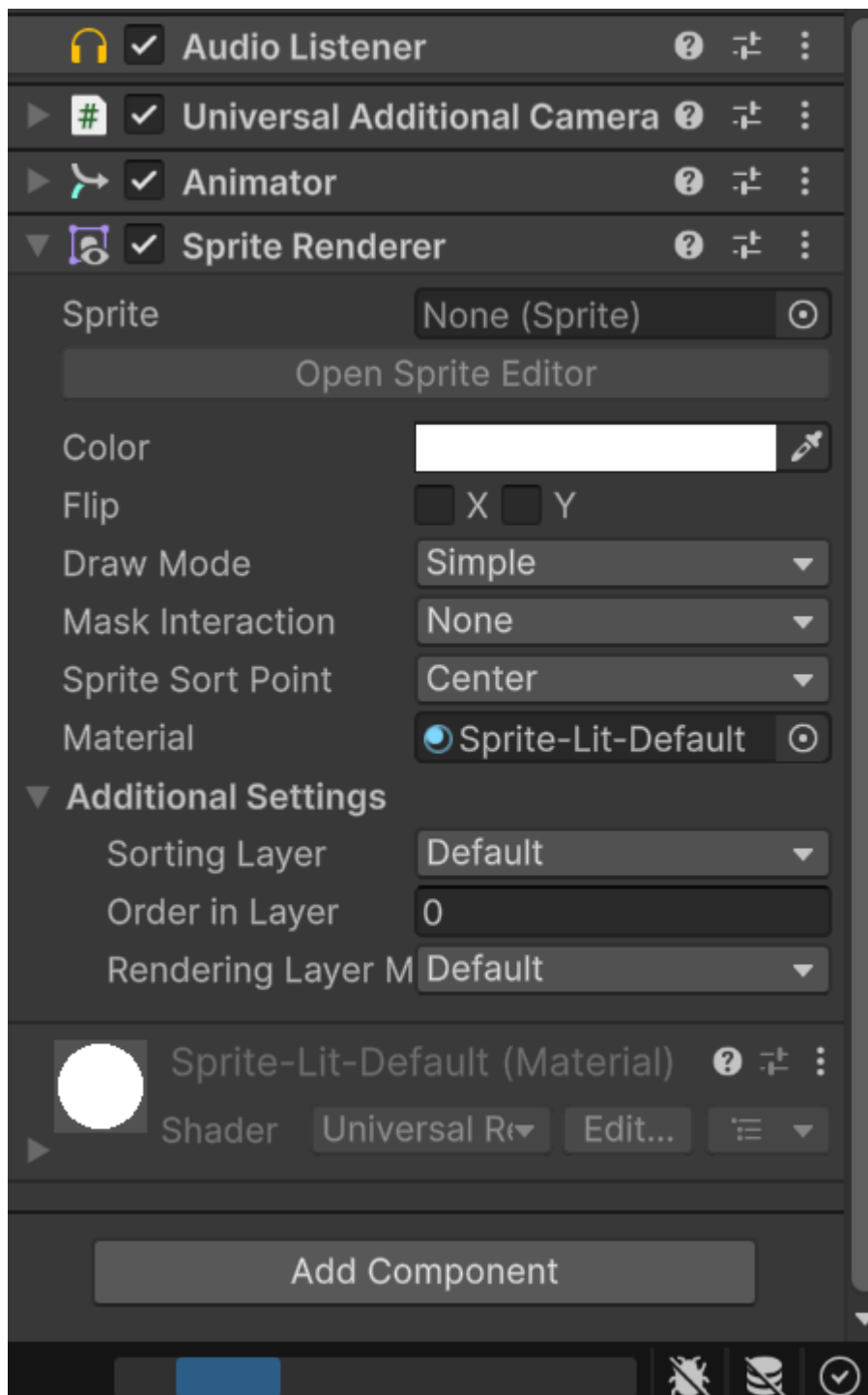
Ser contenedor de otros objetos (para organizar).

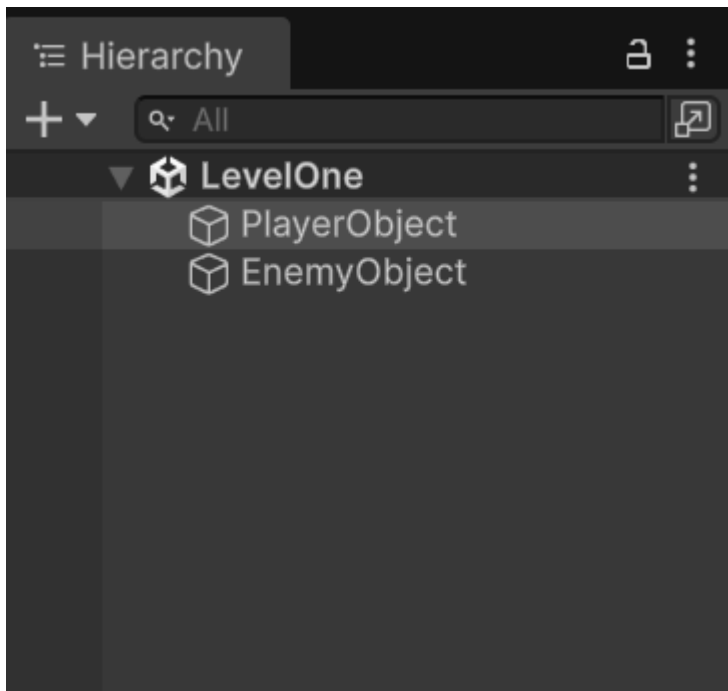
Ser un punto de referencia en el mundo (para generar enemigos, etc.).

Tener solo lógica (añadirle scripts).

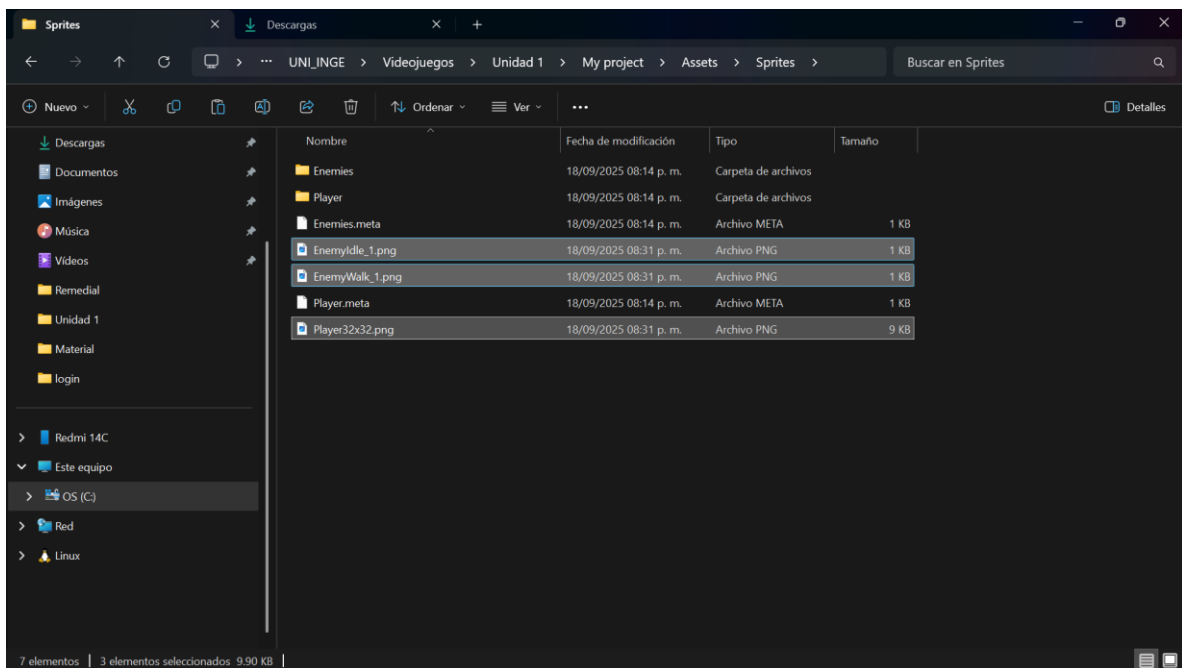
**En transform se pone Posicion y rotation**

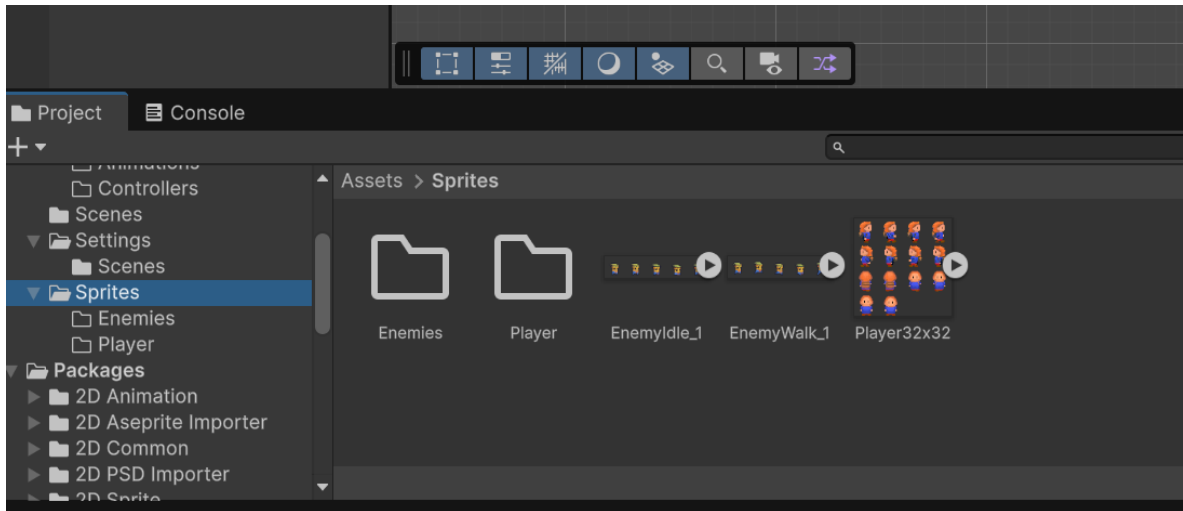


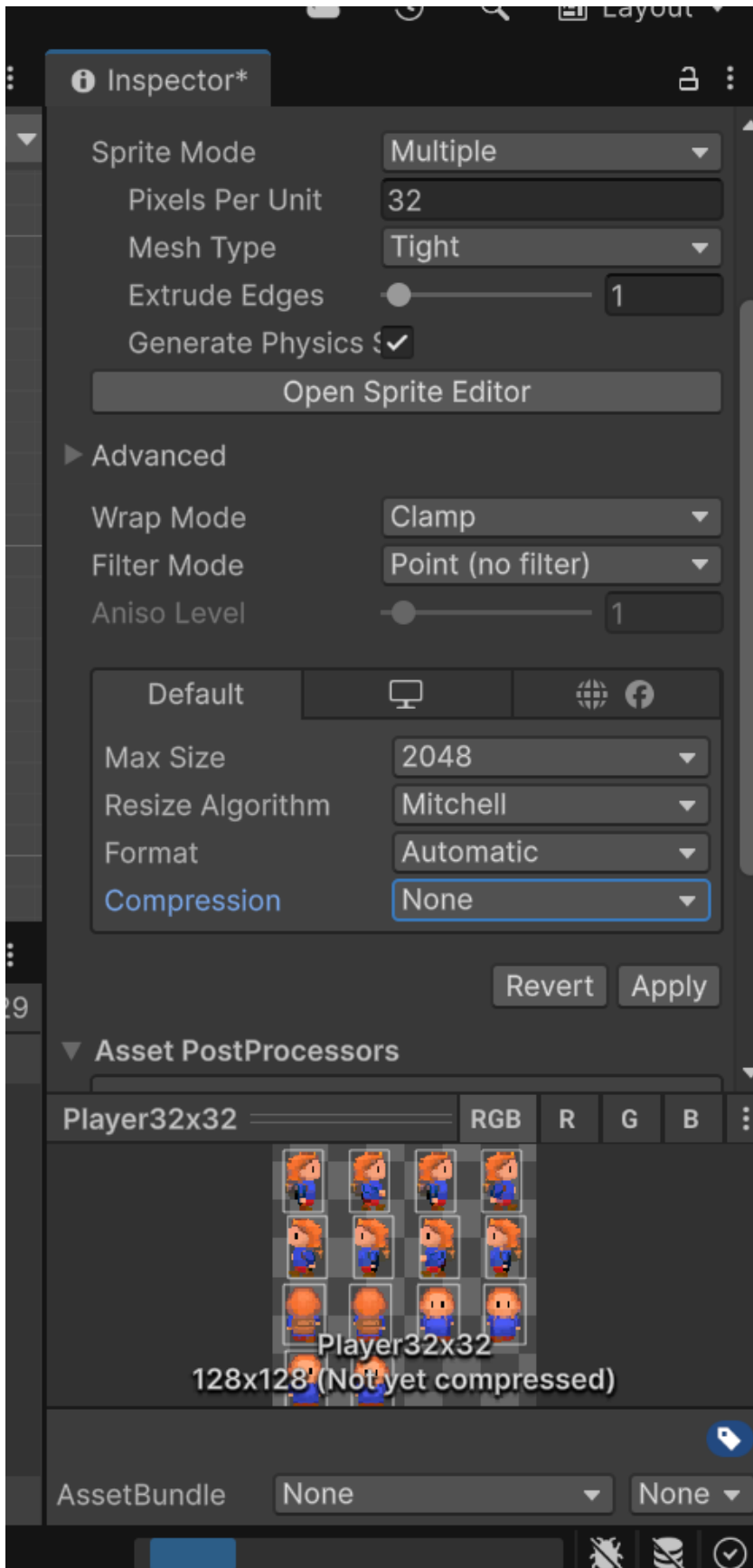




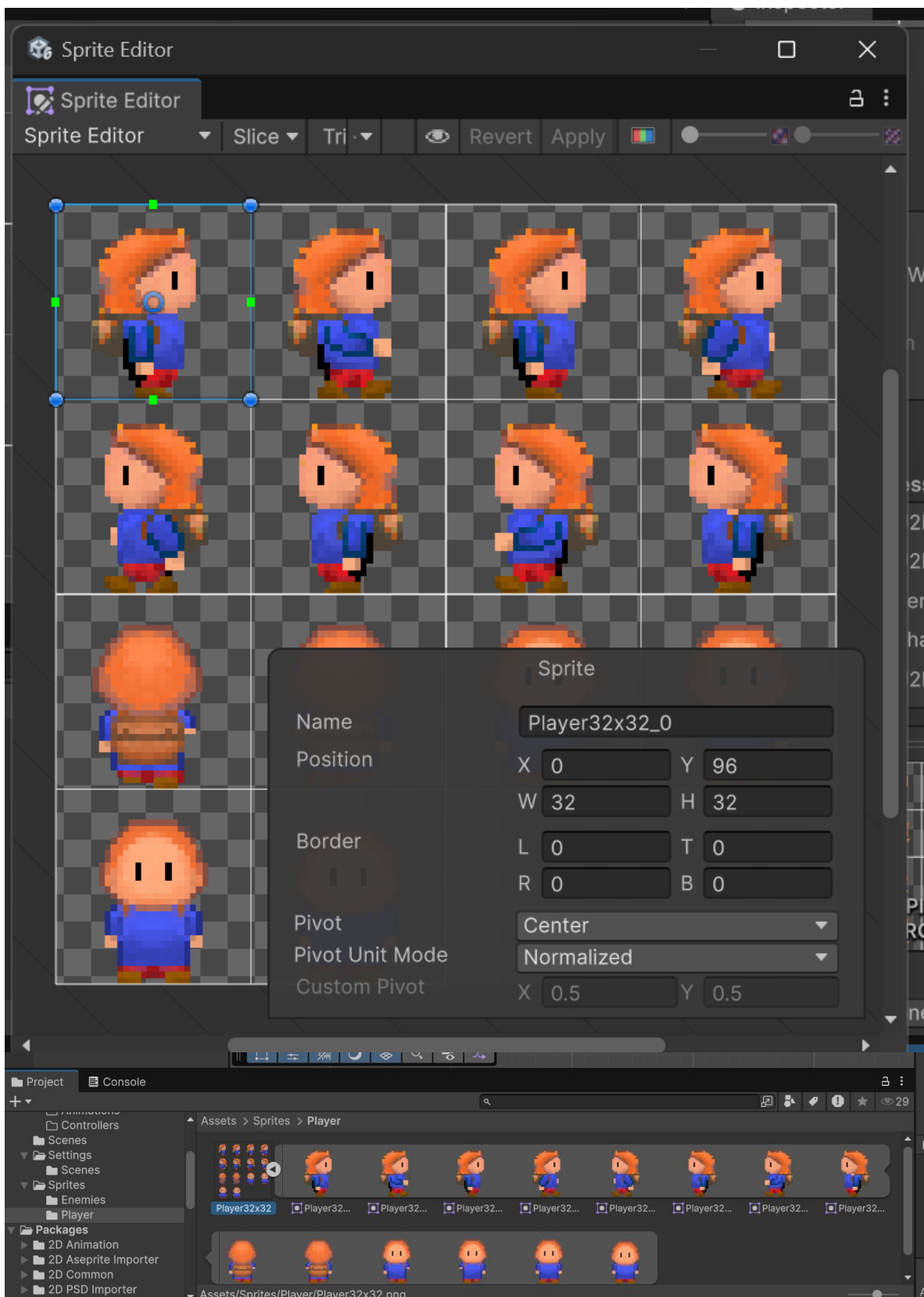
## Se guardan las animaciones



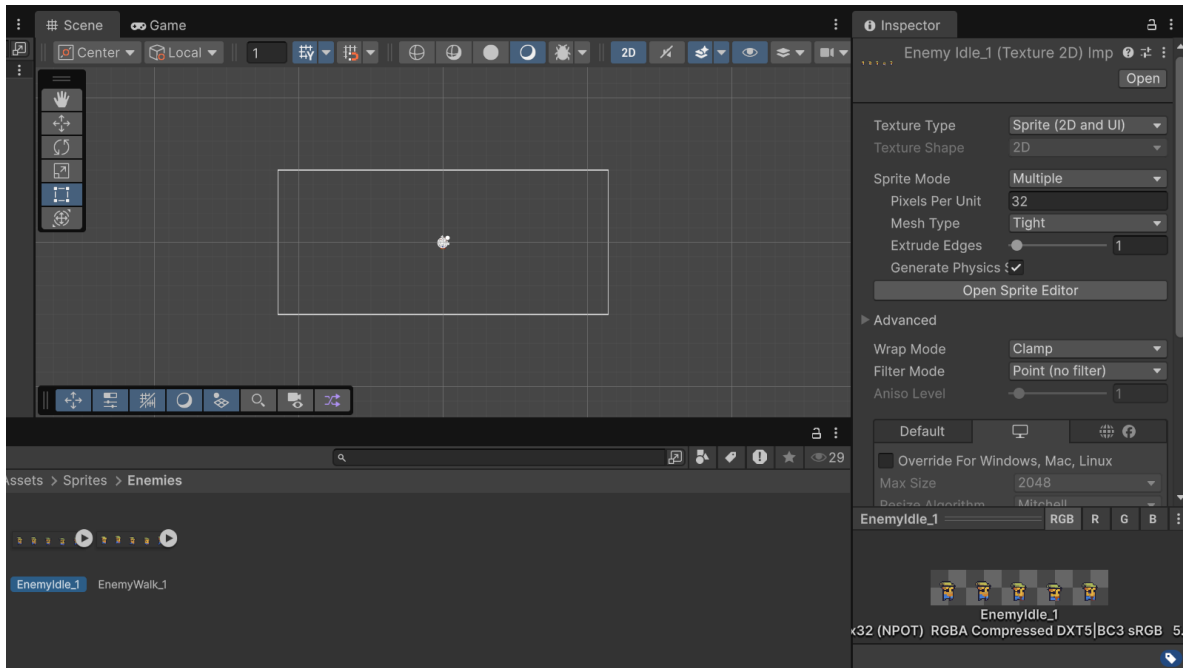




Se organiza el Sprite editor







### 1. ¿Qué es un SpriteSheet y para qué se utiliza en el desarrollo de videojuegos?

Un SpriteSheet es una imagen que contiene múltiples sprites (cuadros de animación o elementos gráficos) organizados en una cuadrícula o disposición específica. Se utiliza para:

Optimizar el rendimiento (menos archivos para cargar).

Facilitar la animación cuadro por cuadro.

Reducir el tamaño total de los recursos del juego.

### 2. ¿Cuál es la ventaja de usar un SpriteSheet en lugar de imágenes individuales?

Rendimiento: Menos llamadas al sistema para cargar texturas.

Organización: Todos los frames de una animación están en un solo archivo.

Memoria: Unity trata el SpriteSheet como una sola textura en GPU, reduciendo el overhead.

Facilidad de animación: Es más sencillo gestionar secuencias animadas.

### 3. ¿Qué tipo de animaciones se pueden crear con un SpriteSheet?

Cualquier animación 2D cuadro por cuadro, como:

Caminar, correr, saltar.

Ataques, habilidades.

Idle (reposo), muerte, interacciones.

Efectos visuales (explosiones, destellos, etc.).

#### **4. ¿Qué formato de archivo es común para SpriteSheets (PNG, JPG, etc.) y por qué?**

PNG es el formato más común.

Porque:

Soporta transparencia (canal alpha).

No tiene pérdida de calidad (compresión sin pérdida).

Es ideal para gráficos pixelados o con bordes definidos.

#### **5. ¿Cómo se relaciona el Sprite Renderer con un SpriteSheet en Unity?**

El Sprite Renderer es el componente que muestra un sprite individual en pantalla.

Unity divide el SpriteSheet en sprites individuales usando el Sprite Editor.

El Sprite Renderer usa uno de esos sprites extraídos (ej: Player\_0, Player\_1) para renderizar el frame actual.

#### **6. ¿Qué herramienta de Unity permite dividir un SpriteSheet en múltiples sprites?**

Sprite Editor (Editor de Sprites).

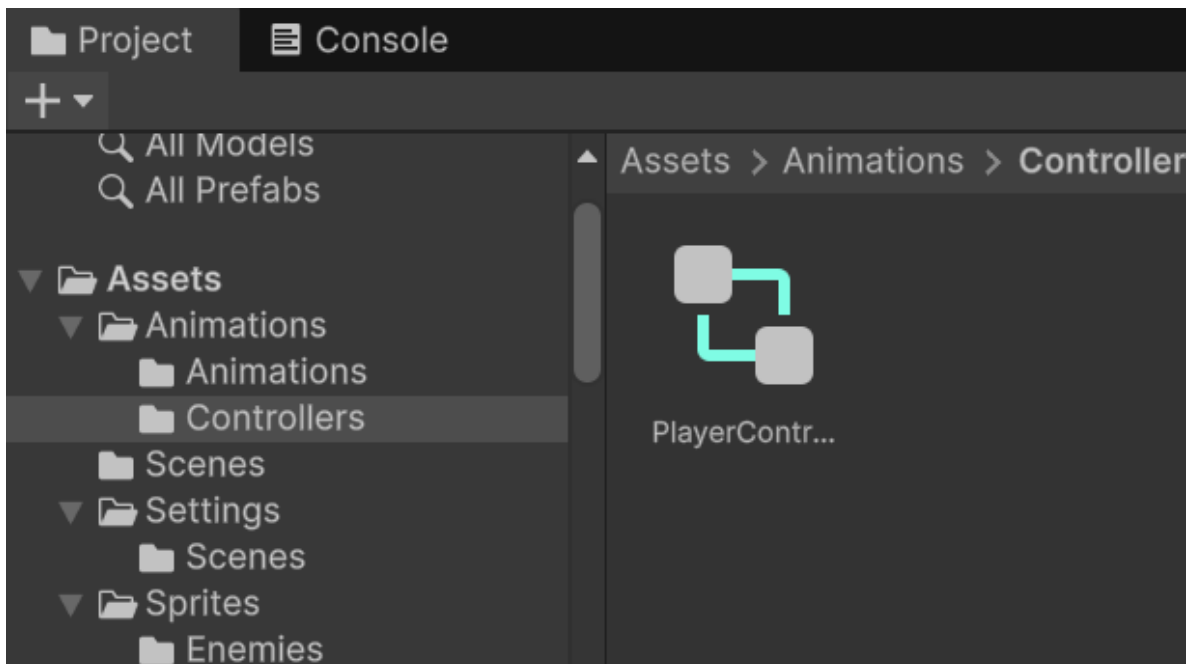
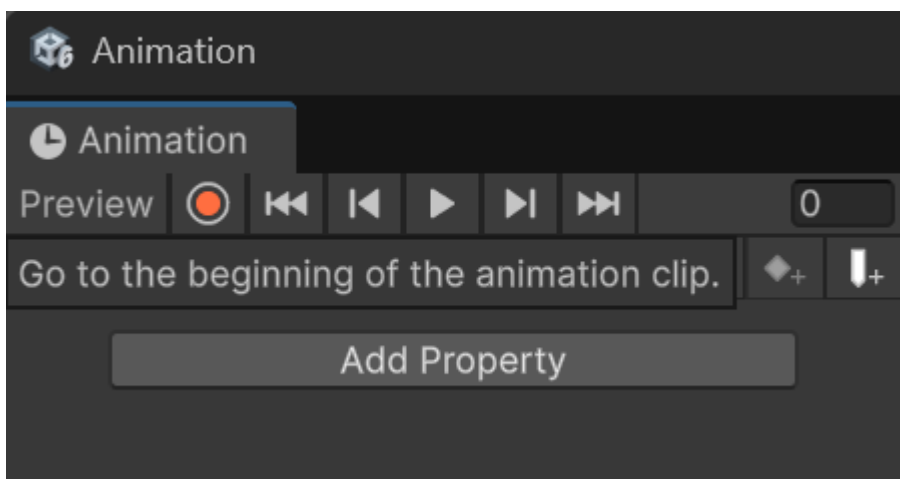
Se accede desde el Inspector al seleccionar un SpriteSheet importado como Sprite (2D and UI) en modo Multiple.

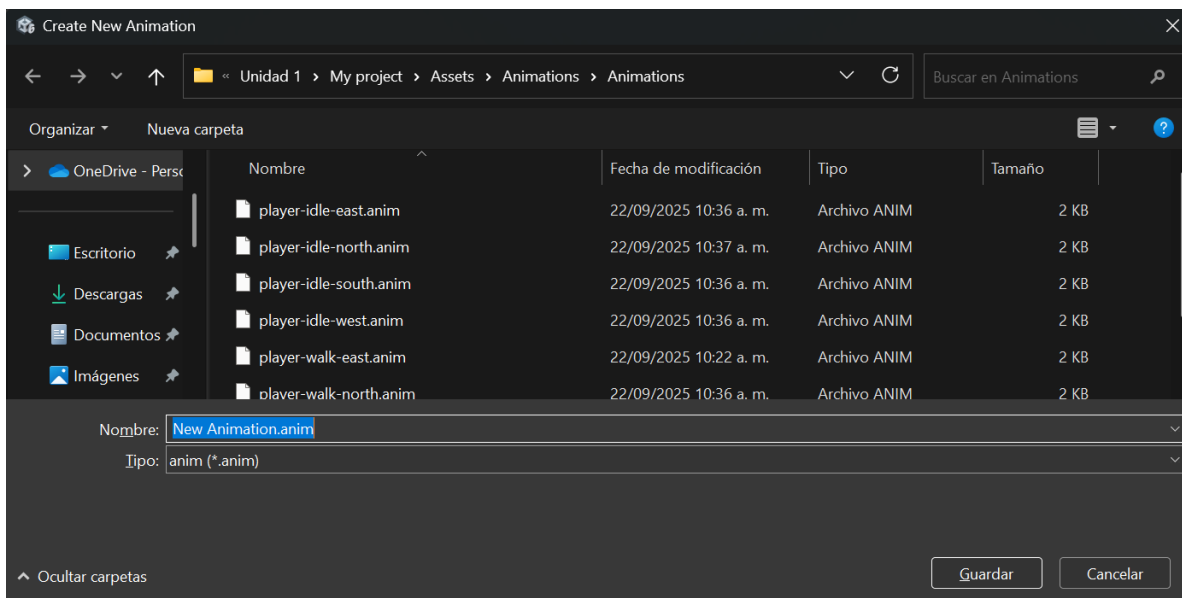
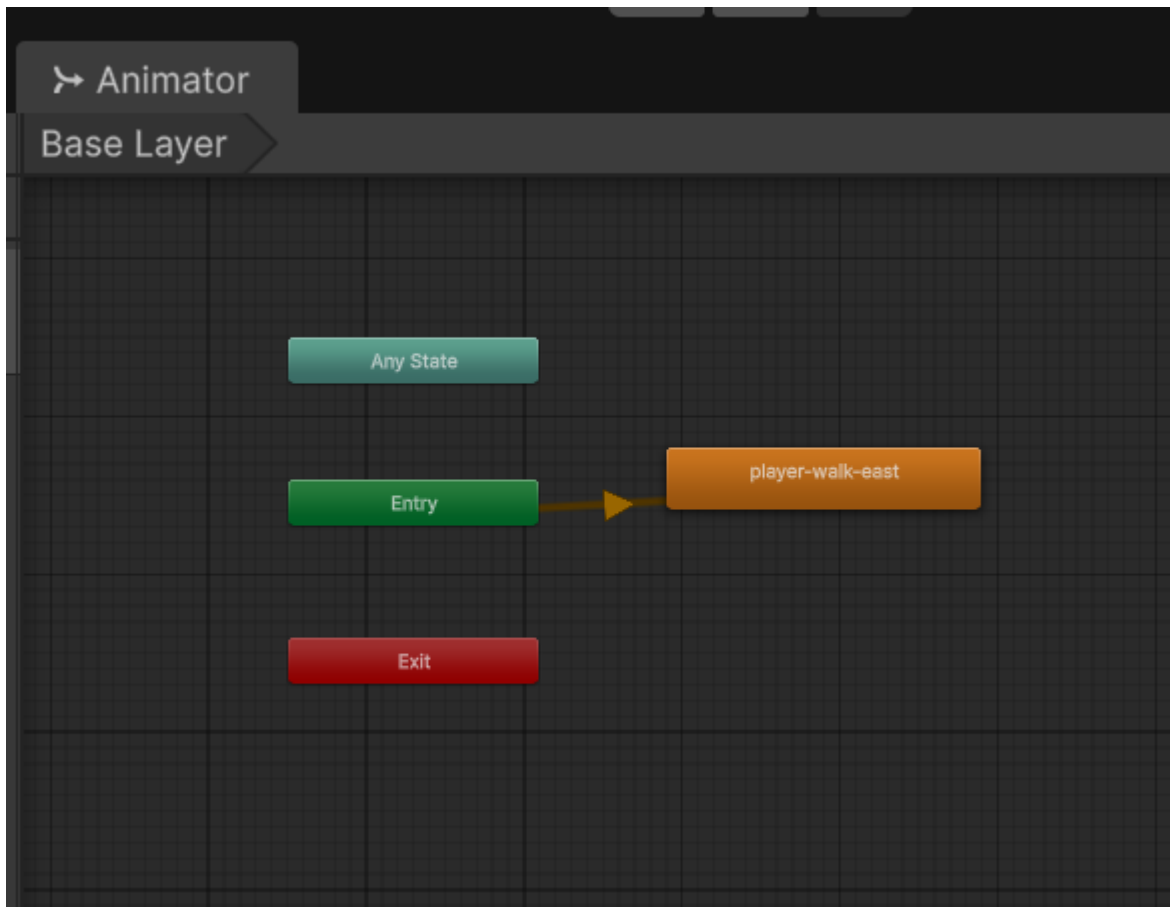
#### **7. ¿Qué parámetros puedes ajustar al importar un SpriteSheet en Unity?**

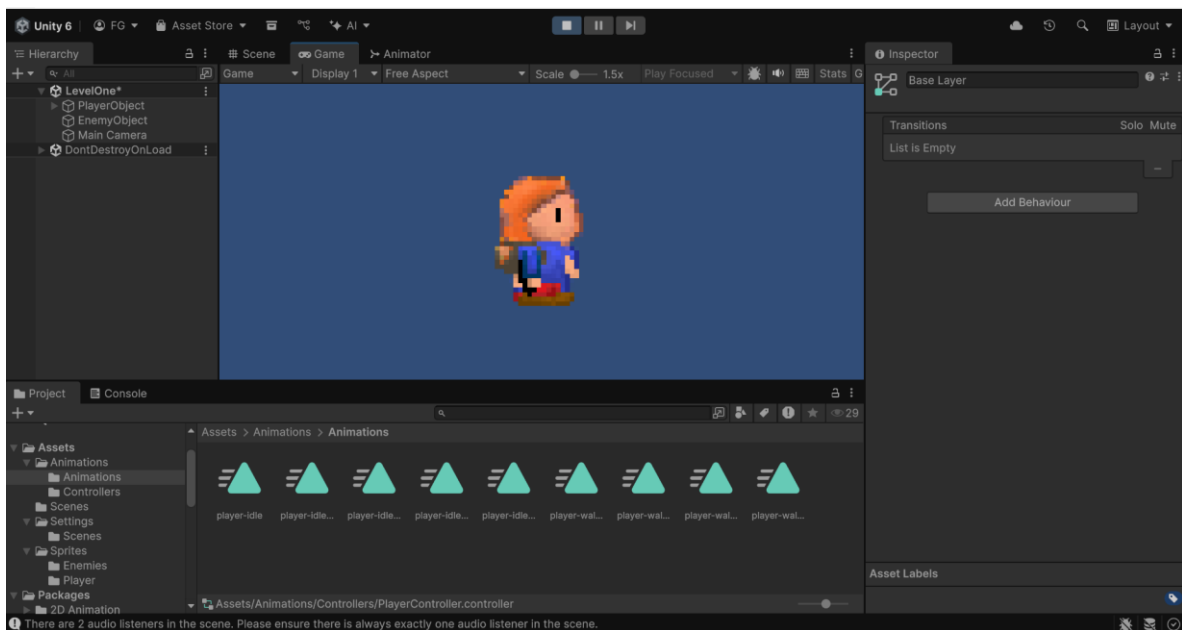
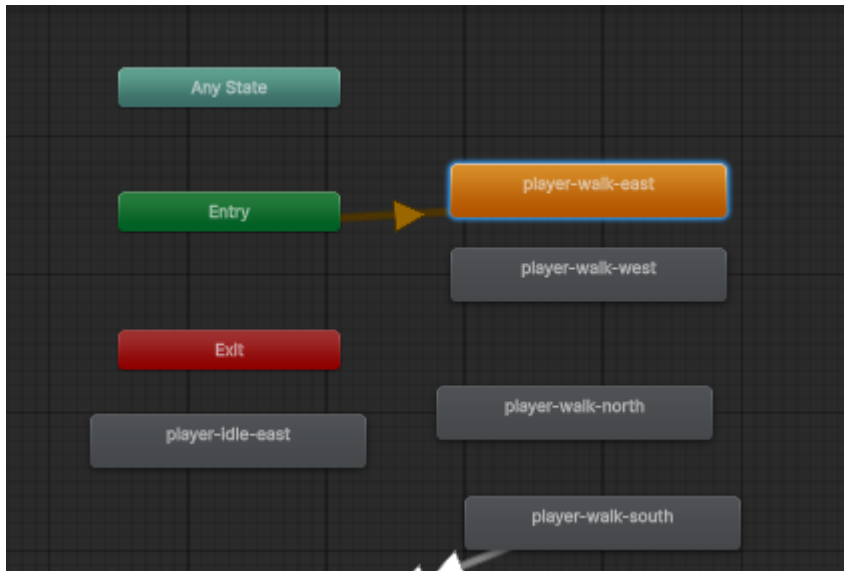
En el Inspector, al importar un SpriteSheet, los parámetros clave son:

Parámetro	Valor típico	Descripción
Texture Type	Sprite (2D and UI)	Define que es un sprite 2D.
Sprite Mode	Multiple	Indica que es una hoja de sprites.
Pixels Per Unit	16, 32, 64	Escala para que coincida con la resolución del juego.
Filter Mode	Point (no filter)	Mantiene píxeles nítidos (ideal para pixel art).
Compression	None o Low	Evita artefactos en sprites pixelados.
Sprite Editor > Slice	Grid By Cell Size	Divide los sprites automáticamente.

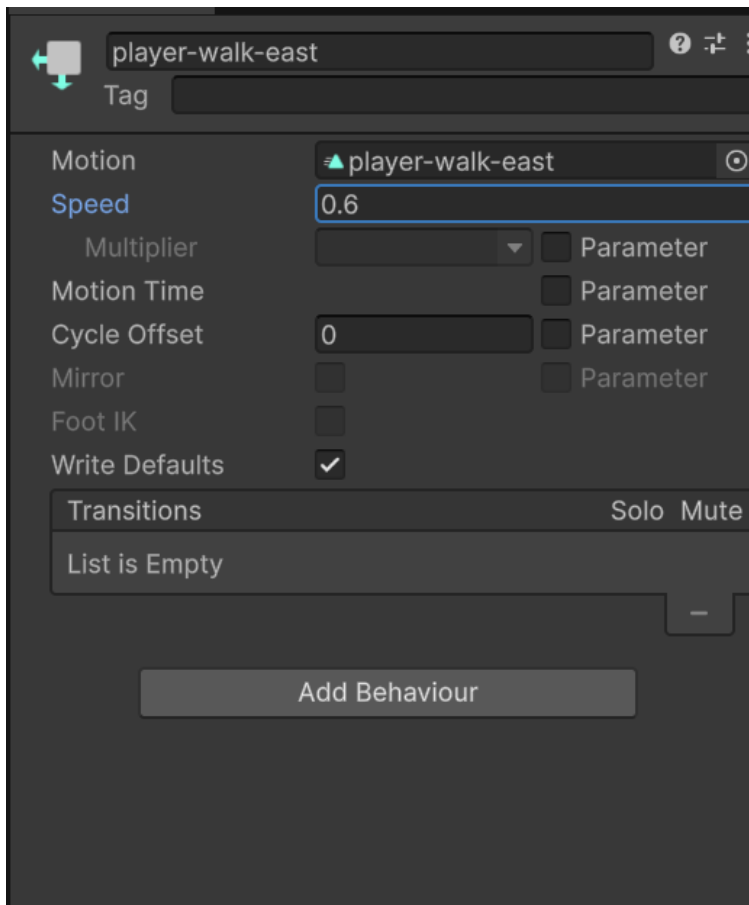
Se le da play







Cambia la velocidad a 0.6



### ¿Qué es una animación en Unity y cómo se representa en el editor?

Una animación en Unity es una secuencia de imágenes o cambios de propiedades que se reproducen en el tiempo para simular movimiento o transformaciones. En el editor, se representa mediante el panel Animation, donde se organizan los fotogramas clave (keyframes) en una línea de tiempo, permitiendo definir cómo varían atributos como la posición, rotación, escala o los sprites de un objeto a lo largo de un intervalo determinado.

### ¿Qué es el componente Animator y para qué se utiliza?

El componente Animator es un elemento que se adjunta a un GameObject para controlar y reproducir animaciones basadas en una máquina de estados. Su función principal es

interpretar las transiciones y parámetros definidos en un Animator Controller, decidiendo qué animación debe ejecutarse en cada momento según las condiciones establecidas (como inputs del usuario o cambios en variables).

### **¿Qué es un Animator Controller y cómo se relaciona con el componente Animator?**

Un Animator Controller es un archivo de asset en Unity que actúa como el cerebro de las animaciones, conteniendo una máquina de estados con sus transiciones, parámetros y reglas lógicas. Se relaciona con el componente Animator siendo asignado a este último, de modo que el Animator ejecuta las animaciones según la configuración y los estados definidos en el Controller.

### **¿Qué función cumple el panel Animator en la gestión de estados?**

El panel Animator permite visualizar y editar la máquina de estados de un Animator Controller, facilitando la creación y conexión de estados, la configuración de transiciones y la definición de parámetros que controlan el flujo de las animaciones. Es la interfaz gráfica donde se diseña el comportamiento animado de un objeto.

### **¿Qué es un "State" y qué representa dentro del Animator?**

Un State (estado) dentro del Animator representa una situación o animación específica en la que puede encontrarse un objeto, como "caminar", "saltar" o "reposo". Cada estado contiene una animación asociada y se conecta con otros estados mediante transiciones, que se activan cuando se cumplen ciertas condiciones, permitiendo un flujo dinámico y coherente entre diferentes acciones animadas.

**Se da play y se ponen las animaciones**

