

Guía de Laboratorio 2

Primera Parte

Mover un personaje con las teclas de dirección en Unity (2D)

Crear una nueva escena

1. Abre Unity y tu proyecto 2D.
2. En el menú superior, haz clic en **File** → **New Scene**.
3. Guarda la escena con el nombre **Level1** (**File** → **Save As**).

Configurar la cámara

1. Selecciona el objeto **Main Camera** en el *Hierarchy*.
2. En el *Inspector*, cambia la propiedad **Projection** a **Orthographic** (si no lo está).
3. Ajusta el valor de **Size** a 3 para acercar la cámara.

Importar y preparar el personaje

1. Arrastra un sprite (imagen del personaje) desde la carpeta **Assets** al *Hierarchy*.
2. Selecciona el objeto creado y cambia su nombre a **Player**.
3. En el *Inspector*, cambia la **Posición** a (0, 0, 0) (Position → X=0, Y=0, Z=0).

Configurar componentes de físicas

1. Con el objeto **Player** seleccionado:
 - Haz clic en **Add Component**.
 - Busca y agrega **Rigidbody 2D**.
 - En el campo **Gravity Scale**, asigna 0, para evitar que el Player caiga.
 - Agrega también un **Box Collider 2D**.
 - Ajusta el tamaño del collider para que encaje con la forma del sprite (usando la opción *Edit Collider*).

Crear el script de movimiento

1. En la carpeta **Assets/Scripts**, haz clic derecho → **Create** → **MonoBehaviour Script**.
2. Nombra el script como Player.
3. Ábrelo y reemplaza el contenido por el siguiente código:

```
using UnityEngine;

public class Player : MonoBehaviour
{
```

```
public float speed = 5;

Rigidbody2D rb2D;

Vector2 movementInput;

void Start()
{
    rb2D = GetComponent<Rigidbody2D>();
}

void Update()
{
    movementInput.x = Input.GetAxisRaw("Horizontal");
    movementInput.y = Input.GetAxisRaw("Vertical");

    movementInput = movementInput.normalized;
}

private void FixedUpdate()
{
    rb2D.linearVelocity = movementInput * speed;
}
}
```

4. Guarda el archivo.
5. Arrastra el script Player al objeto **Player** en el *Hierarchy* para vincularlo.

Solucionar error de configuración del sistema de entrada

En algunas versiones de Unity aparece un error al usar `Input.GetAxisRaw()` porque está activado únicamente el **Input System Package**.

1. Ve a **Edit** → **Project Settings** → **Player**.
2. En la sección **Other Settings** → **Configuration**, busca **Active Input Handling**.
3. Cambia el valor de **Input System Package (New)** a **Both**.
4. Reinicia el editor si es necesario.

Ejecución

1. Presiona el botón **Play**.
2. Usa las flechas del teclado o las teclas **W, A, S, D** para mover al personaje en la escena.

Segunda Parte

Animación del personaje

Cortar los Sprite

Para poder animar un personaje que se encuentra en una hoja de sprites (sprite sheet), primero debemos dividir esa hoja en imágenes individuales.

1. En el **Panel Project**, localiza el archivo **Warrior_Red** (es una hoja de sprites que contiene todas las poses del personaje).
2. Con el sprite seleccionado, ve al **Inspector** y haz clic en el botón **Open Sprite Editor**.
 - Si aparece un mensaje que indica que el Sprite Editor no está instalado, ve a **Window > Package Manager** y asegúrate de que el paquete **2D Sprite** esté instalado.
3. Dentro del Sprite Editor, presiona el botón **Slice** en la parte superior.
4. En la ventana de opciones de Slice, en **Type** selecciona **Grid By Cell Count**.
 - Esto divide la hoja en una cuadrícula según la cantidad de sprites que contiene (6 columnas y 8 filas).
5. Haz clic en **Slice** y luego en **Apply** (esquina superior derecha).
 - Ahora la hoja está dividida en sprites individuales que podrás usar para animar.

Realizar la animación de Inactivo (Idle)

1. En el **Panel Project**, selecciona los sprites **Warrior_Red_0** hasta **Warrior_Red_5**.
 - Para seleccionar en orden, haz clic en el primero, mantén presionada la tecla **Shift**, y haz clic en el último.
2. Arrástralos directamente a la **Scene** o al **Panel Hierarchy**.
3. Unity abrirá automáticamente un cuadro de diálogo para guardar la animación.
 - Dentro de la carpeta **Assets**, crea una nueva carpeta llamada **Animations**.
 - Dentro de **Animations**, crea otra carpeta llamada **PlayerWarriorRed**.
 - Guarda la animación con el nombre **Idle**.
4. Al hacer esto, Unity creará dos cosas automáticamente:
 - El archivo de animación **Idle.anim**.
 - Un **Animator Controller** y un objeto del personaje en la escena (luego lo organizaremos).

Realizar la animación de Caminar (Walk)

1. Repite los pasos anteriores, pero esta vez selecciona los sprites **Warrior_Red_6** hasta **Warrior_Red_11**.
2. Arrástralos a la escena.
3. En el cuadro de diálogo, guarda el archivo de animación en la carpeta **PlayerWarriorRed** con el nombre **Walk**.

Crear el controlador de animación

El controlador organiza y gestiona qué animaciones se usan en el personaje.

1. Elimina los objetos extras que Unity creó automáticamente al arrastrar los sprites a la escena (elimina los dos personajes y sus controladores de animación).
 - Solo nos interesa quedarnos con las animaciones .anim.
2. En la carpeta **PlayerWarriorRed**, haz clic derecho → **Create > Animator Controller**.
3. Cambia el nombre del archivo a **PlayerAnimController**.
4. Selecciona el archivo **PlayerAnimController** y en el **Inspector** haz clic en **Open**.
 - Se abrirá la ventana **Animator**.
5. Arrastra las animaciones **Idle** y **Walk** desde el panel Project hacia la ventana Animator.
 - Esto las vincula al controlador para que puedan reproducirse.

Asignar el controlador al Player

1. Selecciona el objeto **Player** en la escena (el personaje que colocaste en la primera parte de la guía).
2. En el **Inspector**, presiona **Add Component** y busca **Animator**.
3. En la propiedad **Controller** del componente Animator, arrastra el archivo **PlayerAnimController**.
 - Ahora el personaje está vinculado a sus animaciones.

6. Ejecución

1. Presiona el botón **Play** en la parte superior de Unity para ejecutar el juego.
2. Verás que el personaje reproduce la animación **Idle** por defecto.
3. Usa las flechas del teclado (↑, ↓, ←, →) o las teclas **W, A, S, D** para mover el personaje en la escena.
 - Por ahora, aunque el personaje se mueva, siempre mostrará la animación **Idle**.
 - En la **siguiente guía**, programaremos la lógica para cambiar a la animación **Walk** cuando el personaje esté en movimiento.