

# Unity

¡Desarrolla

tu juego!

Animación Idle y Correr  
en Unity



# ¡Desarrolla tu juego!

## Unity

### De Animación Idle y Correr en Unity

#### 1. Introducción

En este capítulo, vamos a hablar de cómo hacer que tu personaje pase de estar más quieto que una estatua a correr como si hubiera visto una oferta limitada de pizzas gratis. Es decir, vamos a implementar dos animaciones básicas pero fundamentales: Idle (quieto) y Run (correr).

#### ¿Qué es una animación Idle?

La animación 'Idle' es la que se reproduce cuando el personaje no está haciendo nada en particular, como esperar instrucciones, reflexionar sobre su existencia o simplemente mirar el horizonte digital.

Aunque parezca simple, una buena animación idle da vida al personaje y hace que el juego se sienta más fluido.



# ¡Desarrolla tu juego!

## Unity

### De Animación Idle y Correr en Unity

#### ¿Y qué pasa con la animación de correr?

La animación de 'Run' o correr se activa cuando el jugador pulsa una tecla para moverse. Es lo que hace que nuestro personaje no patine como si estuviera sobre hielo. Una transición suave entre idle y run mejora mucho la sensación de control.

#### Requisitos

- Tener un personaje con un Animator Controller asociado.
- Tener al menos dos animaciones: Idle y Run.
- Saber cómo usar parámetros dentro del Animator para cambiar entre estados.

#### En resumen

Aunque parezca una tontería, cuidar las animaciones idle y run puede hacer que tu juego se vea más profesional. No se trata solo de moverse, sino de moverse con estilo.



# ¡Desarrolla tu juego!

## Unity

### De Animación Idle y Correr en Unity

#### 1. Crea una carpeta para las animaciones

- Haz clic derecho en el panel Project → Create → Folder → llámala Animations.

#### 2. Crea la animación de correr a la derecha

- Selecciona los sprites de correr a la derecha (por ejemplo: run\_0, run\_1, run\_2...).
- Arrástralos directamente a la escena o al objeto del personaje.
- Unity te pedirá que guardes una animación:
- Llama a la animación CorrerDerecha.anim.
- También te creará automáticamente un Animator Controller, por ejemplo: Personaje.controller.
- Se habrá añadido un componente Animator al personaje con el controller conectado.

#### 3. Abre el Animator

- Ve a Window → Animation → Animator.
- Conecta la animación



# ¡Desarrolla tu juego!

## Unity

### De Animación Idle y Correr en Unity

4. Crea transiciones entre animaciones (necesitas crear parámetro, punto siguiente)

1. Haz clic derecho en CorrerDerecha → Make Transition → conecta con CorrerIzquierda.
2. Haz clic derecho en CorrerIzquierda → Make Transition → conecta con CorrerDerecha.
3. Haz clic en cada flecha de transición y:
  - En Conditions, agrega la condición:
    - Para CorrerDerecha:  $MovimientoX > 0$
    - Para CorrerIzquierda:  $MovimientoX < 0$

4.1 Crea un parámetro para controlar el movimiento

1. En el Animator, haz clic en la pestaña Parameters.
2. Haz clic en el + y elige Float → nómbralo MovimientoX.

5. Código script player

```
//Declaración variables privadas animación: anim y sr  
private Animator anim;  
private SpriteRenderer sr;
```



# ¡Desarrolla tu juego!

## Unity

### De Animación Idle y Correr en Unity

```
void Start() // inicializar las variables de animación en start
{
    anim = GetComponent<Animator>();
    sr = GetComponent<SpriteRenderer>();
}
```

En void Update ()

// Enviar parámetro al Animator: el parámetro llamado MovimientoX

```
anim.SetFloat("MovimientoX", movimiento);
```

// Espejar sprite si no tienes animación específica para izquierda

```
if (movimiento < 0)
```

```
{
```

```
    sr.flipX = true; // Voltea sprite horizontalmente
```

```
}
```

```
else if (movimiento > 0)
```

```
{
```

```
    sr.flipX = false; // Normal
```

```
}
```

