

# JUMPING GUY

Tu primer videojuego 2D completo y multiplataforma en Unity 5



*[Un curso para EscuelaDeVideojuegos.net](http://EscuelaDeVideojuegos.net)*

creado por Héctor Costa

# Bienvenida

- Presentación
- Escuela de Videojuegos
- Etapa GameMaker Studio
- Etapa Unity 5



# Requisitos previos

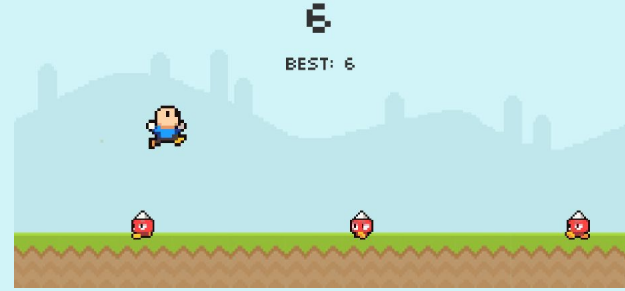
No es un curso para todo el mundo, se requiere:

- Fundamentos de programación orientada a objetos
- Fundamentos de lenguaje C#
- Fundamentos de Unity 5

[En nuestro canal](#) encontrarás una introducción gratuita a C# y un repaso por los fundamentos de Unity, si eres principiante empieza por ahí ;-)

# Temario

- Diseño multiplataforma
- Efecto parallax en distintas capas
- Animaciones, colisiones y partículas
- Generación dinámica de objetos
- Manipulación del ritmo de juego
- Manejo de la GUI y marcador de puntos
- Guardado de puntuación máxima con PlayerPrefs
- Exportación a Windows, WebGL y Android



**¡Te espero en la siguiente lección y no  
olvides suscribirte a nuestro canal!**

**[escueladevideojuegos.net](http://escueladevideojuegos.net)**

# 0) Recursos

El curso está pensado para que te centres en aprender

- En los adjuntos del vídeo encontrarás todos los recursos, un PDF con las diapositivas y otro PDF con un resumen completo del curso

¡Úsalos para apoyar tu aprendizaje!

# 1) Diseño multiplataforma



¿Cómo logramos que un juego se vea bien en el máximo de dispositivos posible?

# Relación de aspecto

Cálculo que relaciona el ancho y alto de una imagen

Relación	Valor	Descripción
<u>5/4</u>	<u>1.25</u>	<u>Estándar en monitores de ordenador prácticamente cuadrados.</u>
<u>4/3</u>	1.33	Estándar del formato PAL para televisores, pantallas de ordenador y iPads.
<u>3/2</u>	1.5	Estándar del formato NTSC para televisores y algunos smartphones.
<u>16/9</u>	<u>1.77</u>	<u>Estándar de pantallas HD, conocido como formato panorámico o widescreen.</u>

**5:4**

**4:3**

**3:2**

**16:9**



# Resolución base

¿Cómo elegir una? ¿Juego pixel art? ¿Buscamos excelencia?

Nombre	Resolución
Full HD	1920 x 1080
HD+	1600 x 900
HD	1280 x 720
qHD	960 x 540
<u>nHD</u>	<u>640 x 360</u>

*Resoluciones 16/9*

**Resolución  
Base**

# Zona segura

Espacio del escenario visible en cualquier dispositivo

*Ancho = Alto \* Relación aspecto mínima*

Resolución	Zona Segura
1920 x 1080	1350 x 1080
1600 x 900	1125 x 900
1280 x 720	900 x 720
960 x 540	675 x 540
<u>640 x 360</u>	<u>480 x 360</u>

*Resoluciones 16/9*



# Diseñando nuestro juego

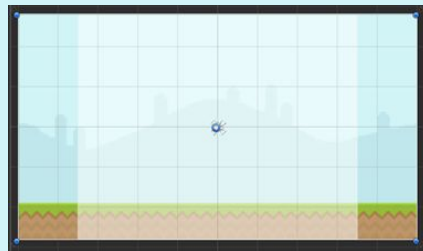
Todos los elementos importantes dentro de la zona segura



## 2) Creando la escena

¿Qué vamos a hacer?

- Crear el proyecto y configurar la escena
- Maquetar el diseño en un Canvas con Raw Images
- Limitar la zona segura con un Panel



Objetivo

- Preparar el escenario para el efecto parallax

### 3) Efecto Parallax de fondo

¿Qué vamos a hacer?

```
public float parallaxSpeed;  
public RawImage background;  
public RawImage platform;
```

- Preparar imágenes como texturas
- La propiedad UV Rect de las Raw Images
- A modificar esta propiedad desde un script

Objetivo

- Conseguir un doble efecto de movimiento en el fondo

## 4) Portada animada e inicio

¿Qué vamos a hacer?



- Crear varios estados de juego
- Crear una UI con un título y descripción animada
- Controlar el efecto Parallax y esconder la UI

Objetivo

- Iniciar el juego al presionar una tecla o botón

# 5) Creando al protagonista

¿Qué vamos a hacer?

- Preparar los sprites del personaje
- Crear un objeto contenedor y un cuerpo
- Posicionarlo y darle el tamaño correcto

Objetivo

- Tener listo nuestro personaje para añadir animaciones



## 6) Animación de correr



¿Qué vamos a hacer?

- Añadir dos animaciones utilizando herencia de objetos
- Controlar los cambios de animación desde scripts
- Enviar mensajes a objetos para ejecutar sus métodos

Objetivo

- Que el personaje esté quieto y al iniciar el juego corra



# 7) Animación de saltar

¿Qué vamos a hacer?

- Añadir una nueva animación
- Controlar la animación desde scripts
- Manejar las transiciones entre los estados

Objetivo

- Que el personaje pueda saltar



## 8) Creando al enemigo



¿Qué vamos a hacer?

- Añadir un nuevo objeto posicionado con herencia
- Escalar correctamente el tamaño de su sprite
- Crear su animación inicial y otorgarle movimiento

Objetivo

- Crear un enemigo animado que se mueva a la izquierda

## 9) Autodestruir enemigos



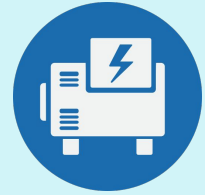
¿Qué vamos a hacer?

- Añadir un objeto que destruya enemigos
- Crear las colisiones entre el destructor y los enemigos
- Utilizar un TAG para identificar objetos en las colisiones

Objetivo

- Liberar la memoria de los enemigos fuera de la escena

# 10) Generador de enemigos



¿Qué vamos a hacer?

- Crear un objeto para generar enemigos en una posición
- Instanciar el prefab del enemigo desde un script
- Programar la función InvokeRepeating

Objetivo

- Crear un objeto que genere un enemigo cada X tiempo

# 11) Animación de muerte



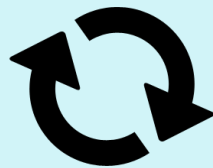
¿Qué vamos a hacer?

- Crear la animación y la colisión contra los enemigos
- Configurar el nuevo estado en el personaje
- Cancelar el generador de enemigos

Objetivo

- Que el personaje muera al chocar contra un enemigo

# 12) Reinicio de juego



¿Qué vamos a hacer?

- Crear un método para reiniciar la escena
- Detectar la muerte del personaje y llamar al método
- Refactorizar algo de código

Objetivo

- Permitir al jugador reiniciar el juego después de morir

# EXTRA: Bug Android



¿Qué vamos a hacer?

- Crear un nuevo estado ready (preparado)
- Añadir un método para cambiar a este estado
- Llamarlo en el evento de animación al morir

Objetivo

- Evitar que el jugador reinicie el juego sin esperarse

# 12) Música y sonidos



¿Qué vamos a hacer?

- Crear un AudioSource para reproducir la melodía
- Crear otro AudioSource para reproducir los efectos
- Gestionar todo el audio desde scripts

Objetivo

- Configurar la música y los efectos de sonido del juego



# 13) Dificultad progresiva



¿Qué vamos a hacer?

- Introducir la propiedad `Time.timeScale`
- Crear un método de incremento y otro de reinicio
- Iniciar y cancelar la invocación del método incremental

Objetivo

- Aumentar la dificultad del juego progresivamente

# 14) Partículas al correr

¿Qué vamos a hacer?

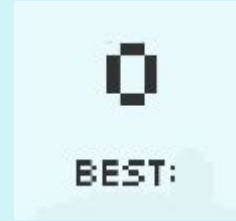
- Crear un sistema de partículas
- Adaptarlo para que parezca polvo al correr
- Manejar las partículas desde el script del jugador

Objetivo

- Generar un efecto de levantar tierra al correr



# 15) Marcador de puntos



¿Qué vamos a hacer?

- Duplicar la capa UI Idle y crear otra para el marcador
- Hacer que aparezca cuando empiece el juego
- Crear un trigger en los enemigos para manejar puntos

Objetivo

- Mostrar marcador e incrementarlo al saltar enemigos

# 16) Guardar récord con PlayerPrefs

¿Qué vamos a hacer?



- Introducir el uso de las funciones PlayerPrefs
- Crear métodos para guardar y consultar un PlayerPrefs
- Controlar los puntos, guardar el récord y mostrarlo en UI

Objetivo

- Mostrar el récord y guardarlo en un fichero

# 17) Exportación multiplataforma

¿Qué vamos a hacer?

- Configurar y exportar el juego para Windows
- Configurar y exportar el juego para WebGL
- Configurar y exportar el juego para Android



Objetivo

- Conseguir los ejecutables para cada plataforma

***Mucho más en***  
***[EscuelaDeVideojuegos.net](http://EscuelaDeVideojuegos.net)***

***¡Hasta pronto!***

***Un curso creado por Héctor Costa***

