

Taller de Godot:  
**Crear un  
Videojuego  
Shoot'em Up**

Cristian Ramanzin



# ◆ El camino recorrido



Desde 2009  
desarrollando web

# ◆ El camino recorrido



Desde 2019  
desarrollando  
videojuegos en **Koru**

# ARKANET

The image features the title 'ARKANET' in a bold, yellow, serif font with a black outline and a slight 3D effect. The text is centered horizontally and is flanked by two clusters of interlocking yellow gears. The background is a dark, atmospheric scene with a cloudy sky in shades of brown and grey. In the foreground, there are dark, jagged rock formations and several large, dark gears, some of which are partially visible and appear to be part of a larger mechanical structure. The overall aesthetic is steampunk and industrial.



**Veamos el resultado**





# Tus primeros pasos con Godot



Curso de  
**Documento de  
Diseño de  
Juego**

Lucio Negrello

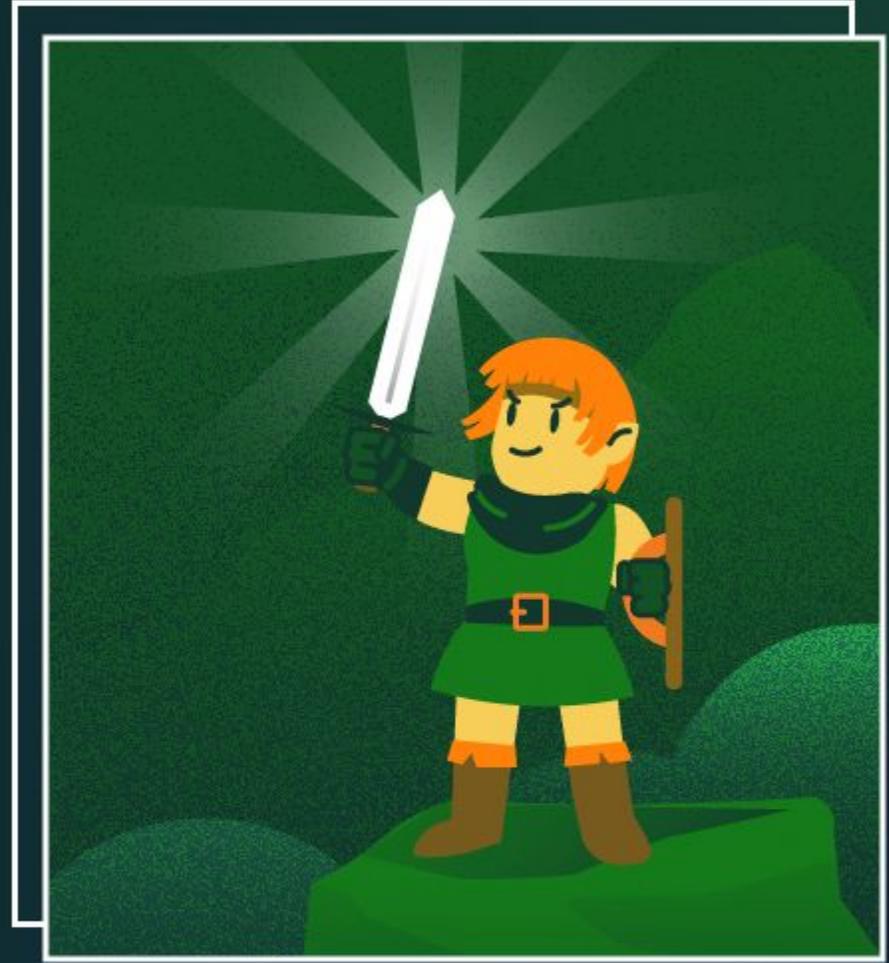
Curso de  
**Concept Art**  
para Videojuegos

Hernán Novillo



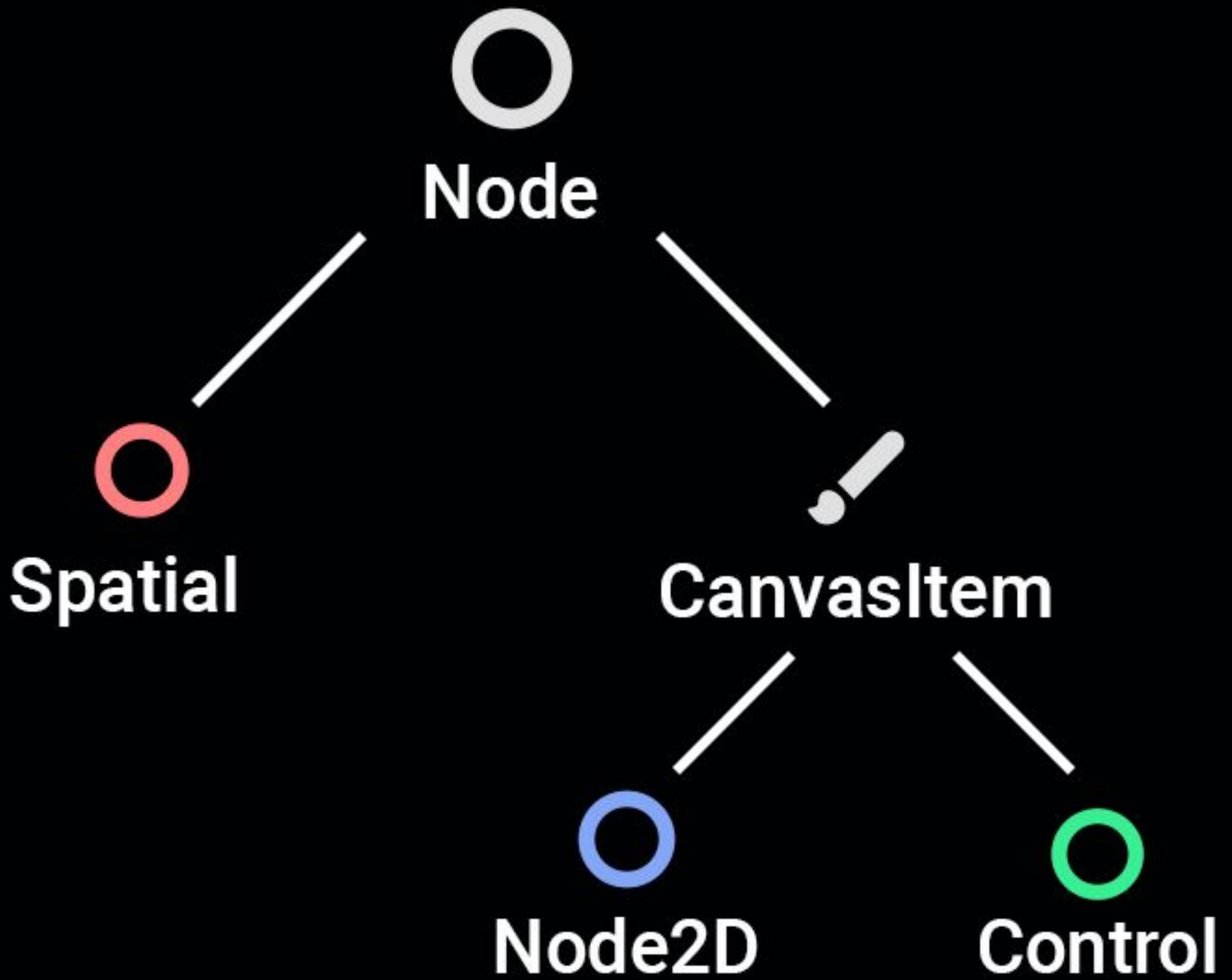
Curso de  
**Artbook para  
videojuegos**

Hernán Novillo





# Las físicas y los elementos en Godot



# KinematicBody2D



Su movimiento depende del jugador y no del motor de físicas



# RigidBody2D



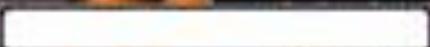
Su movimiento depende del motor de físicas

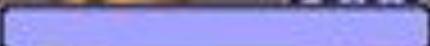
# Bazooka

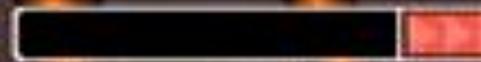
tor  
00



Mr. Lardtree  
100

Human Team 1   h

★ Soldados  



# StaticBody2D



Es un cuerpo que no se mueve.



# Area2D



Es una región limitada por nosotros donde detectamos quien entra y quién sale.

# ◆ Personaje principal





**¿Cuál crees que es el  
mejor nodo para  
nuestro personaje?**



Creando  
el personaje  
principal

# ◆ Pausa

— **KinematicBody2D:** Es un cuerpo que se mueve con código y no se ve afectado por las físicas del juego.



# Animaciones

## ◆ Reto

**— Crea una animación en el AnimationPlayer modificando las propiedades del KinematicBody2D para usar más adelante cuando el personaje reciba daño.**



Creando  
tu primer enemigo

## ◆ Reto

**Diseña tus propios patrones de movimientos para los grupos de enemigos.**



Movimiento  
dirigido al jugador

# ◆ Cortinilla

— **Atributo position:** Es la posición en la que se encuentra el nodo en relación con el nodo padre. (Slide 2)

— **Atributo global\_position:** Es la posición del nodo en relación a la pantalla. (Slide 3)



Creando  
el disparo

# ◆ Cortinilla

**preload:** Esta función te permite cargar un recurso que se encuentra en una dirección estática. Si estos recursos son escenas, los podemos instanciar y agregar a nuestro juego por medio de código.

## ◆ Reto

**Teniendo en cuenta donde estamos instanciando la escena disparo, haz que las balas dejen de seguir al personaje.**



# Colisiones

## ◆ Reto

— Crear función de `set_expllosion()` en el enemigo que se mueve en grupo.



Cuando el jugador  
recibe daño



# Usando singletons



Creando PowerUp

## ◆ Reto

**Crea tu propio power ups, modificando cantidad de balas, direcciones y/o velocidad.**



Creando un fondo  
en loop



# Optimizando movimientos y memoria

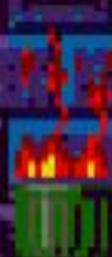


# Creando el HUD

SCORE 29600

TIME 0:39

RINGS 3



 SONIC  
x 3

## ◆ Reto

**Usando la función que controla las vidas en el singleton, reinicia el puntaje del jugador a 0 cuando pierde**



# TileMap



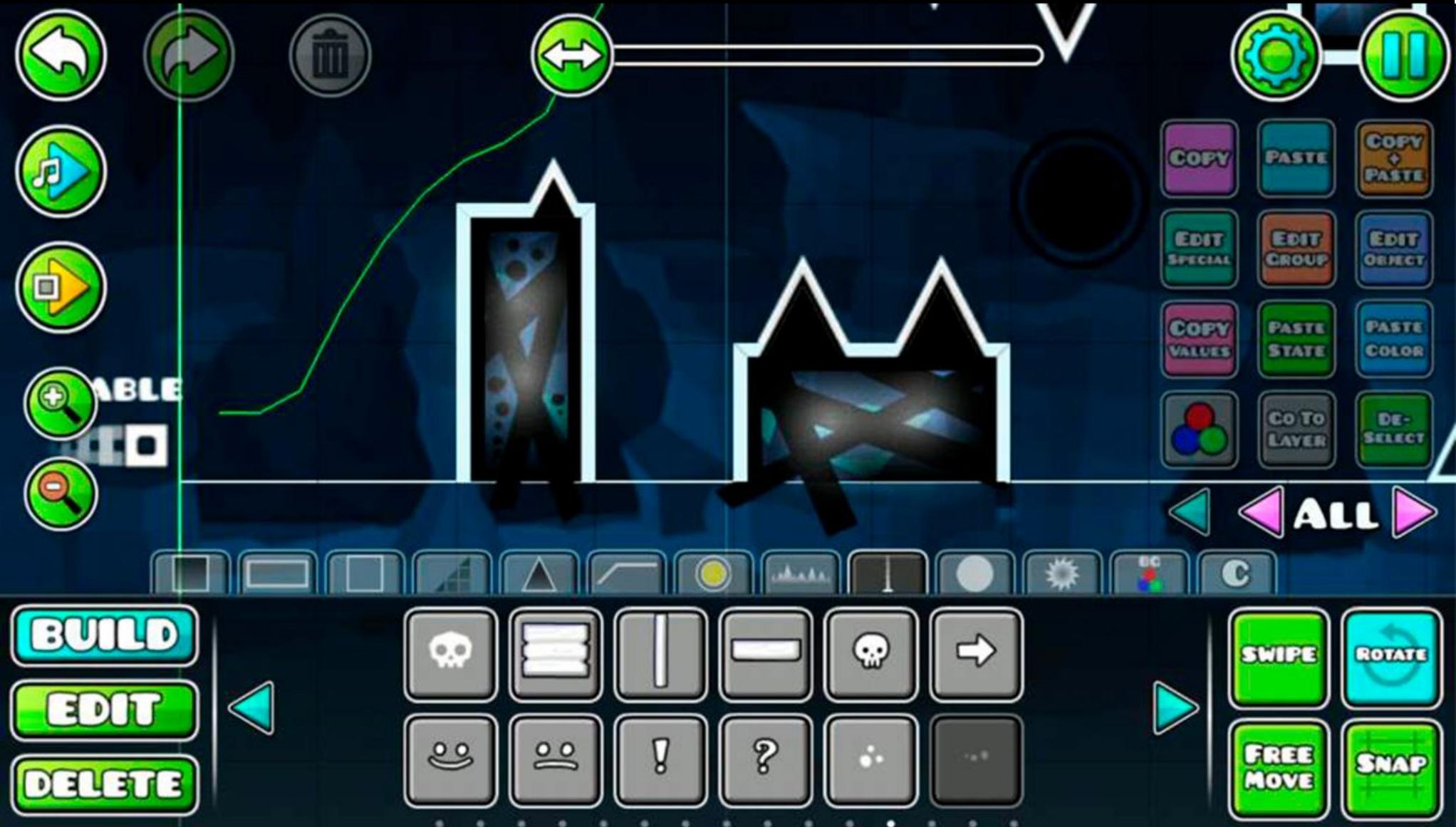
# Cinemática de la meta



Diseñando  
nuestro primer  
nivel



# GEOMETRY DASH



# GRAVITY ACE

NEW OPEN IMPORT EXPORT CAMPAIGN LEVEL STATS RESTART PLAY HELP

SELECT (S) ADD (SPACE) BRUSH (B) ERASE (E) UNDO (Z) REDO (Y)

- 
- 
- 
- 
- 
- 
- 
- 



186 -50  
[OPEN FOLDER](#) ::

# ◆ Cortinilla

- Boceta en papel para generar ideas.
- Prueba la mayor cantidad posible de combinaciones de los mismos elementos a lo largo de los niveles.
- Luego de una situación difícil, crea un descanso para el jugador
- Enséñale al jugador con los elementos del juego



# Diseño de otros niveles



# Asignando SFX y música del juego



Creando  
pantalla principal



Exportar tu juego



¡Lo lograste!