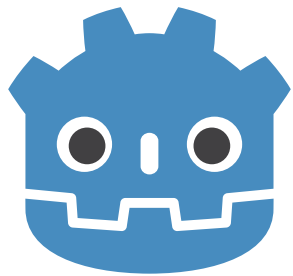


Introducción al desarrollo de videojuegos 2D con Godot Engine (Parte II)

Webinar – 02 de Abril de 2020



GODOT



IEEE

Computational Intelligence Society

**Games Technical
Committee**

Temario

- Estudio de caso práctico (Arkanoid Godot):
 - ✓ *Sonidos.*
 - ✓ *Áreas2D.*
 - ✓ *Árbol de Escenas.*
 - ✓ *Condición de victoria y derrota.*
 - ✓ *Interfaz.*
 - ✓ *Niveles.*

Repositorio de recursos

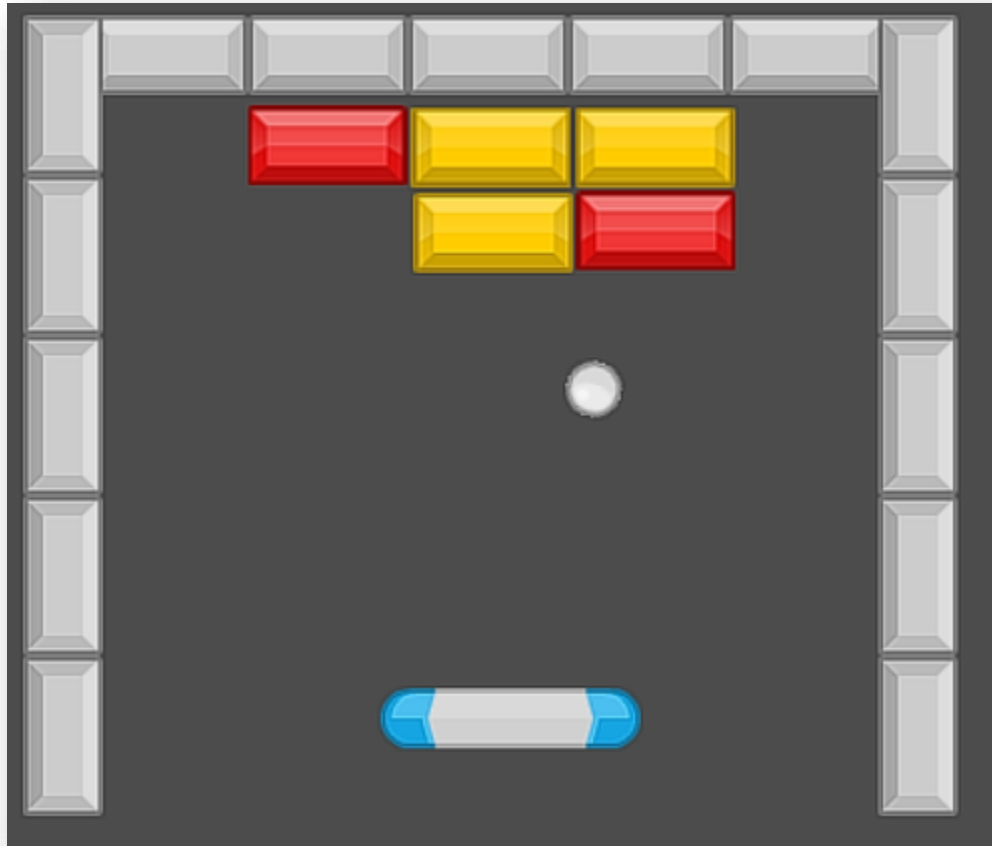
Disponible en GitHub:

- Presentación (Parte I y II).
- Proyecto de caso práctico inicial (Parte I y II).
- Proyecto de caso práctico final (Parte I y II).
- Assets.



<https://github.com/gastoncaminiti/WebinarGodotIEEEGTC2020>

Estudio de caso práctico: Arkanoid Godot



Realizaremos a continuación el desarrollo de elementos del videojuego Arkanoid partiendo del proyecto inicial en el repositorio.

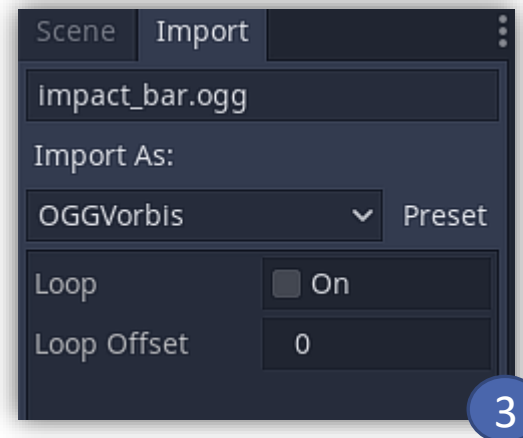
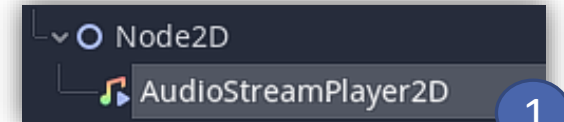
`projects/Parte 2/Arkanoid Godot (Inicio)`



Usando nodos de audio

1. Crear un nodo **AudioStreamPlayer2D**.
2. Importar sonidos en Godot Engine.
Formato admitido: .ogg y .wav
3. Revisar configuración del asset importado.
4. Realizar llamada ***play()*** por código.

```
if body.is_in_group("bar"):
>I $Ball.apply_central_impulse(Vector2(0, -100))
>I $Audio_Barra.play()
```



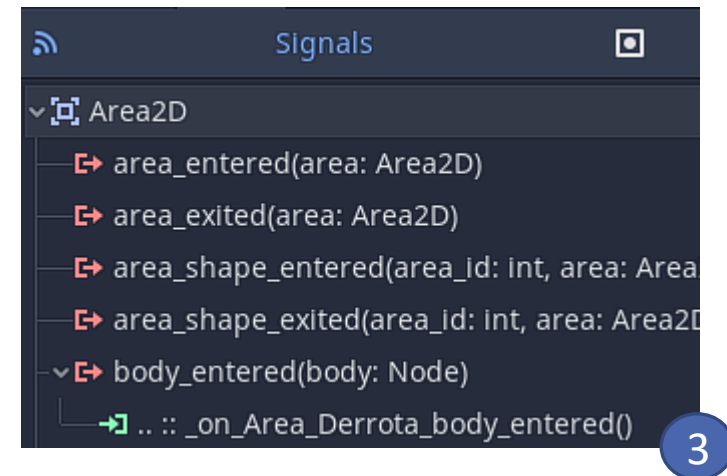
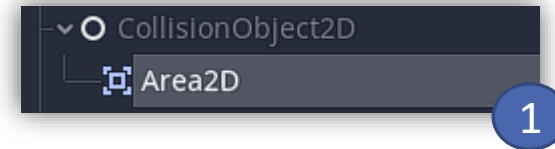
4

3

Usando nodos de Area2D

1. Crear un nodo **Area2D**.
2. Crear nodo hijo **CollisionShape2D**.
3. Conectar señal **body_entered()** con el padre.
4. Definir por código la acción al detectar cuerpo.

```
25 func _on_Area_Derrota_body_entered(_body):  
26     >| # Condición de derrota: Pelota en zona por debajo de la barra.  
27     >| $GUI.ver_mensaje_lose()
```



4

Usando funciones del árbol de escenas

1. Para determinar cantidad de nodos existentes de un grupo.
2. Para reiniciar la escena.
3. Para pausar la escena.
4. Para cargar una nueva escena.

```
get_tree().get_nodes_in_group("bar-deleted").size()
```

1

```
get_tree().paused = true
```

3

```
get_tree().reload_current_scene()
```

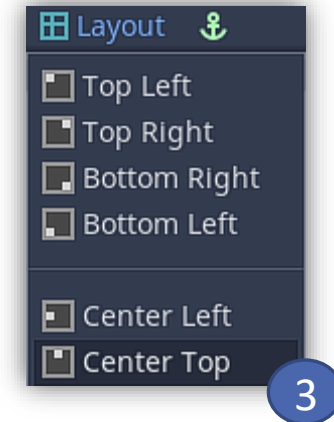
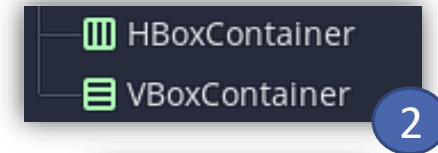
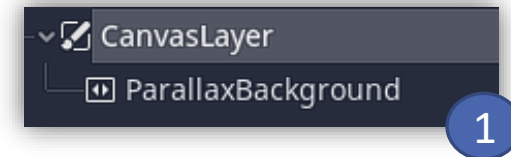
2

```
get_tree().change_scene("res://escenas/level2.tscn")
```

4

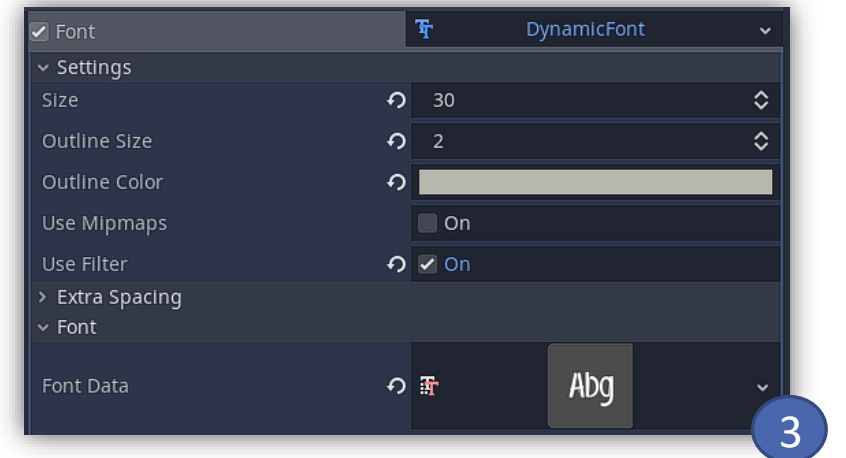
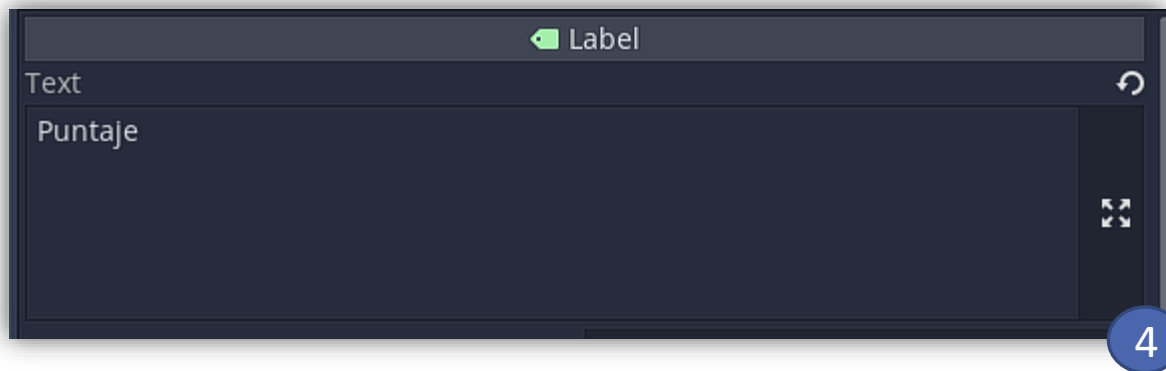
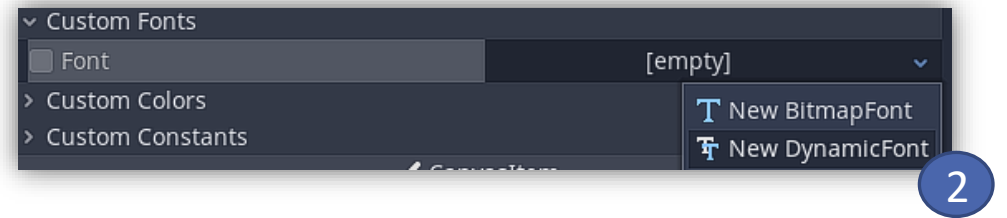
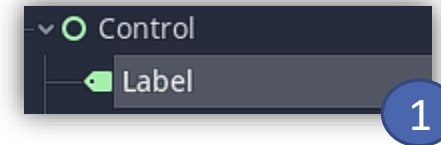
Creando interfaces: Capas

1. Crear una escena con nodo **CanvasLayer**.
2. Crear un nodo de control **VBoxContainer**.
3. Alinear el nodo al centro de la capa.
4. Crear un nodo de control **HBoxContainer**.



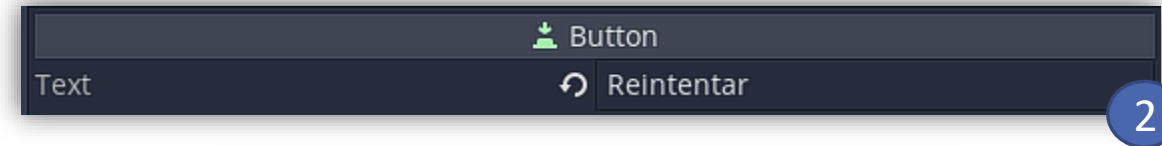
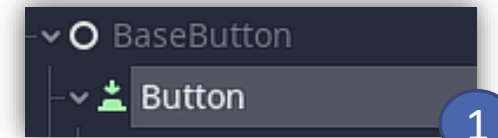
Creando interfaces: Label

1. Crear un nodo hijo **Label**.
2. Asignar un nuevo **DynamicFont**.
3. Asignar una nuevo **FontData** (.ttf)
4. Definir texto a visualizar.

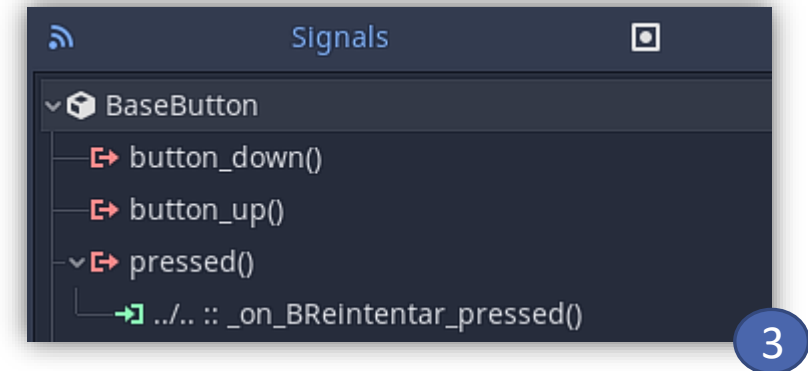


Creando interfaces: Botón

1. Crear un nodo hijo **Button**.
2. Definir texto a visualizar.
3. Conectar padre con señal **pressed()**.
4. Definir por código la acción al presionar.



```
14 func _on_BReintentar_pressed():  
15     >| get_tree().reload_current_scene()  
16
```



4

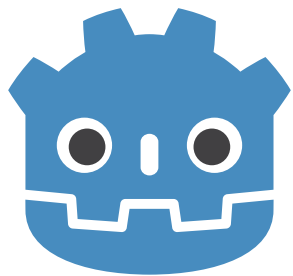
Recursos adicionales

- [Documentación de Godot Engine \(Español\)](#)
- [Godot Engine Series](#)
- [Zamna Atlante – Open 2D Platform game en Godot 3.1](#)
- [Verdandi Norse Legends](#)
- [IEEE GTC Argentina](#)
- [Godot regional communities team](#)



¡Gracias!
#quedatenecasa

Prof. Gastón Caminiti. CC by 4.0



GODOT



IEEE

Computational Intelligence Society

**Games Technical
Committee**