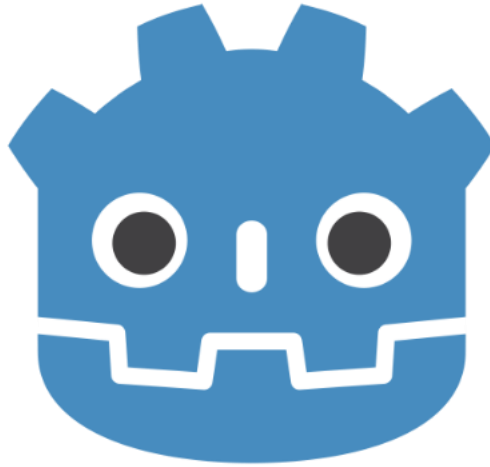


# DOCUMENTACION PROYECTO GODOT



## Índice

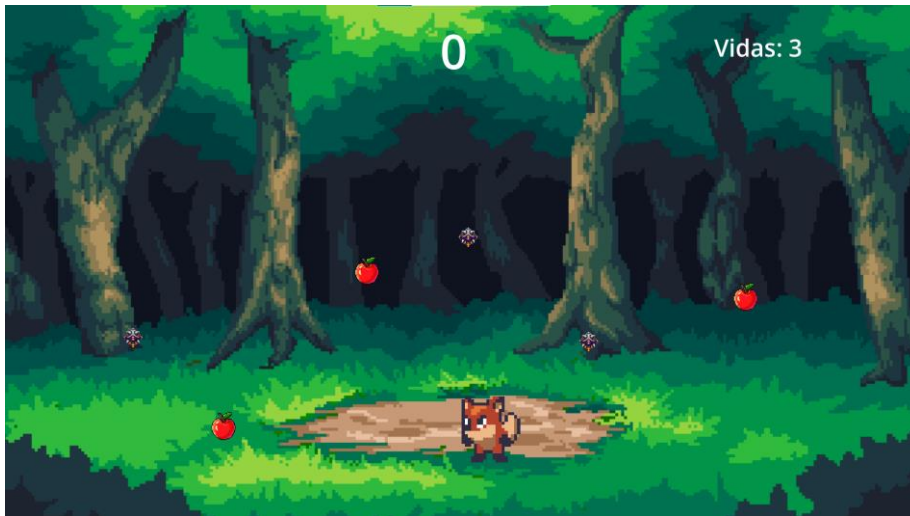
1. ✈ Introducción
2. 🗂 Estructura del Proyecto
3. ⚙ Mecánicas del Juego
4. 📄 Implementación en Godot
5. 📁 Scripts y Código
  - Jugador (jugador.gd)
  - Spawner (spawner.gd)
  - Controles (controles.gd)
  - Pantalla de Muerte (pantalla\_muerte.gd)
  - Menú Principal (menu.gd)
  - Sistema Global (global.gd)
6. 📱 Controles Móviles
7. 🎵 Sonidos y Animaciones
8. 📁 Configuración del Proyecto
9. ⚡ Posibles Mejoras
10. 🌀 Assets Cogidos

## 1.INTRODUCCION

**Atrapa Manzanas** es un juego 2D desarrollado en **Godot** , en el que el jugador controla un personaje que debe atrapar manzanas mientras evita obstáculos. El juego contiene:

- Controles para **PC y dispositivos móviles**.

- **Diferentes niveles de dificultad.**
- Un **sistema de vidas.**
- **Sonidos y animaciones** para mejorar la experiencia de juego.



## 2. ESTRUCTURA DEL PROYECTO.

### ■ Escenas (res://Escenas/)

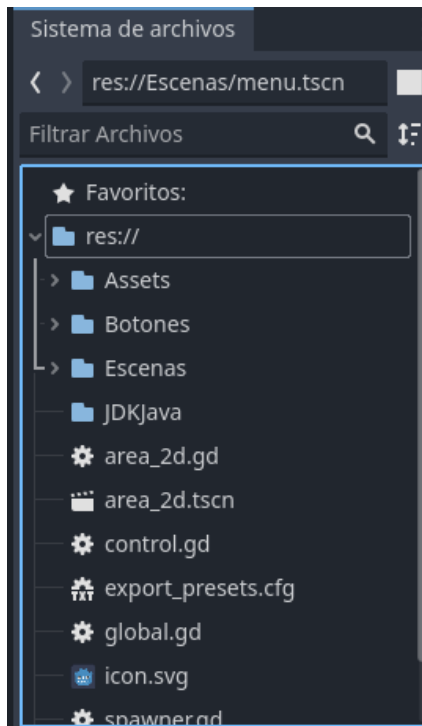
- menu.tscn → Menú principal del juego.
- juego.tscn → Escena principal del juego.
- jugador.tscn → Personaje principal.
- pantalla\_muerte.tscn → Pantalla de Game Over.
- controles.tscn → Controles táctiles para móviles.

### ■ Scripts (res://Scripts/)

- jugador.gd → Controla el movimiento del personaje.
- spawner.gd → Genera las manzanas y obstáculos.
- pantalla\_muerte.gd → Maneja la pantalla de Game Over.
- menu.gd → Configura la selección de niveles.
- controles.gd → Implementa los controles táctiles.
- global.gd → Sistema global para la dificultad.

### ■ Assets (res://Assets/)

- manzana.png, bosque.png, spike.png → Sprites del juego.
- cogermanzana.mp3, morir.mp3 → Sonidos.



### 3. MECÁNICAS DEL JUEGO

- Movimiento del Jugador:** Puede moverse a la izquierda/derecha y saltar.
- **Sistema de Vidas:** El jugador tiene **3 vidas**. Si pierde todas, aparece la **Pantalla de Game Over**.
- **Dificultad:** Se elige al inicio del juego (Fácil, Medio, Difícil).
- Controles Móviles:** En Android/iOS, aparecen botones táctiles para moverse y saltar.
- Sonidos:**
  - cogermanzana.mp3 → Suena al atrapar una manzana.
  - morir.mp3 → Suena cuando se acaban las vidas.

## 4.IMPLEMENTACION EN GODOT

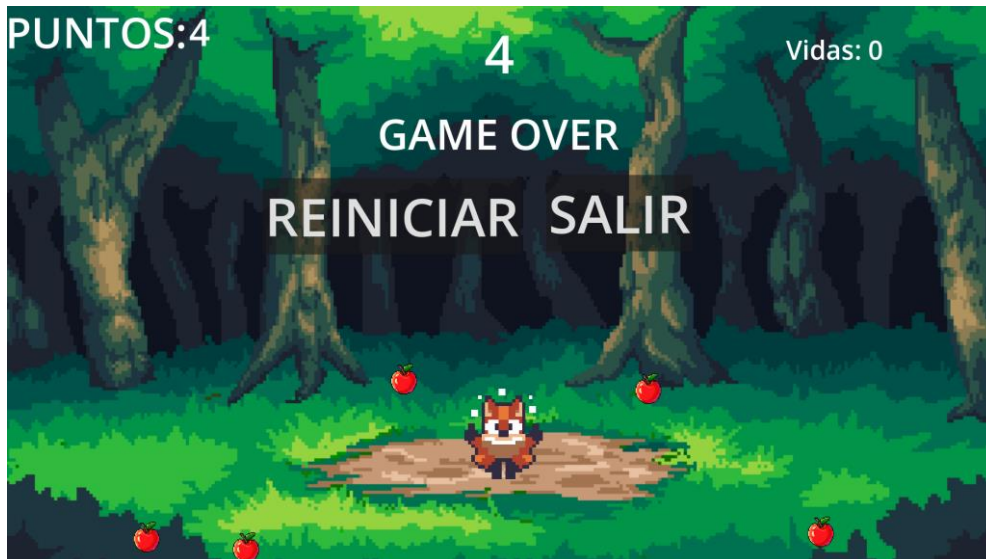
-Menú Principal (menu.tscn) → Permite seleccionar el nivel de dificultad o salir del juego.

-Juego (juego.tscn) → Carga al personaje, spawners y HUD.

-Spawner (spawner.tscn) → Genera manzanas y trampas aleatoriamente.

-Jugador (jugador.tscn) → Controla el personaje y su interacción con el entorno.

-Pantalla de Game Over (pantalla\_muerte.tscn) → **Muestra opciones para reiniciar o volver al menú.**



## 5. SCRIPTS Y CÓDIGO

### 1. Jugador (jugador.gd)

-Gestiona el **movimiento**, las **vidas**, los **controles táctiles/teclado** y los **sonidos**.

```
extends CharacterBody2D
```

```
var speed := 500
```

```
signal scoreUP
```

```
signal playerDied
```

```
var jump_force := -500
```

```
var gravity := 20
```

```
var vidas := 3
```

```
@onready var animated_sprite_2d = $AnimatedSprite2D
```

```
@onready var vidas_label = $"../Vidas"
```

```
@onready var sonido_manzana = $cogermanzana
```

```
@onready var sonido_morir = $morir
```

```
func _ready():
```

```
    actualizar_vidas() # Se actualizan las vidas en pantalla al inicio
```

```
func _physics_process(delta):
```

```
    # Verifica tanto teclado como botones táctiles
```

```
    var inputvel = 0
```

```
    if Input.is_action_pressed("ui_left") or Input.is_action_pressed("izquierda"):
```

```
        inputvel = -1
```

```

elif Input.is_action_pressed("ui_right") or Input.is_action_pressed("derecha"):

    inputvel = 1

velocity.x = inputvel * speed

# Manejo de gravedad

if not is_on_floor():

    velocity.y += gravity

# Detección de salto (teclado o botón táctil)

if (Input.is_action_just_pressed("ui_accept") or
Input.is_action_just_pressed("saltar")) and is_on_floor():

    velocity.y = jump_force

move_and_slide()

# ♦ Animaciones

if velocity.x != 0:

    animated_sprite_2d.play("correr")

    animated_sprite_2d.flip_h = velocity.x < 0

else:

    animated_sprite_2d.play("inactivo")

func subirScore():

    emit_signal("scoreUP")

    if sonido_manzana:

        $cogermanzana.play() # Reproducir sonido al coger una manzana

```

```

func morirse():

    vidas -= 1

    actualizar_vidas() # Actualizar el texto de vidas

    if vidas > 0:

        print("⚠ Perdiste una vida. Vidas restantes:", vidas)

        posicion_inicial() # Se reinicia el personaje en su posición inicial

    else:

        print(" Se acabaron las vidas. Game Over.")

        animated_sprite_2d.play("herido")

        # Reproducir sonido de muerte

        if sonido_morir:

            $morir.play()

        emit_signal("playerDied")

        await get_tree().create_timer(0.5).timeout

        get_tree().call_group("UI", "mostrarPantallaMuerte")

func actualizar_vidas():

    if vidas_label:

        vidas_label.text = "Vidas: " + str(vidas) # Muestra las vidas en pantalla

    else:

        print("⚠ Error: No se encontró VidasLabel")

```

```
func posicion_inicial():
```

```
    position = Vector2(100, 300) # Ajusta según la posición inicial del jugador
```

## Spawner (spawner.gd)

-Maneja la generación de **manzanas y trampas** en la escena.

```
extends Node2D
```

```
var Coleccionable = preload("res://area_2d.tscn") # Carga la escena de la  
manzana/bomba
```

```
func _ready():
```

```
    # Configurar el temporizador para generar objetos periódicamente
```

```
    $Timer.stop()
```

```
    $Timer.wait_time = randf_range(2, 4)
```

```
    $Timer.start()
```

```
func _on_timer_timeout():
```

```
    # Asegurar que estamos usando la dificultad seleccionada
```

```
    var dificultad_actual = Global.dificultad
```

```
    # Incrementar la velocidad con el tiempo o dificultad
```

```
    Global.velocidad_manzanas += 5 * dificultad_actual
```

```
    # Crear una nueva instancia del objeto Coleccionable
```

```
    var coleccionable = Coleccionable.instantiate()
```



```

# Ajustar su velocidad según la dificultad actual

coleccionable.velocidad = Global.velocidad_manzanas # Asegurar que los
objetos usen la velocidad global

# Agregar el objeto a la escena

add_child(coleccionable)

print("Nueva manzana/bomba generada con velocidad:",
Global.velocidad_manzanas)

```

## Controles (controles.gd)

-Muestra u oculta los botones táctiles según la **plataforma**.

```
extends CanvasLayer
```

```
@onready var botones = $"."
```

```
func _ready():
```

```
    # Detectar si el juego se ejecuta en un móvil (Android/iOS)
```

```
    if OS.get_name() == "Android" or OS.get_name() == "iOS":
```

```
        print(" 📱 Jugando en un dispositivo móvil: Mostrando botones")
```

```
        botones.show() # Mostrar controles táctiles
```

```
    else:
```

```
        print(" 💻 Jugando en PC: Ocultando botones")
```

```
        botones.hide() # Ocultar controles en PC
```

```
func _on_derecha_pressed():
```

```
    $derecha.modulate.a = 0.5
```

```
func _on_derecha_released():  
    $derecha.modulate.a = 1
```

```
func _on_izquierda_pressed():  
    $izquierda.modulate.a = 0.5
```

```
func _on_izquierda_released():  
    $izquierda.modulate.a = 1
```

```
func _on_saltar_pressed():  
    $saltar.modulate.a = 0.5
```

```
func _on_saltar_released():  
    $saltar.modulate.a = 1
```

## Pantalla de Muerte (pantalla\_muerte.gd)

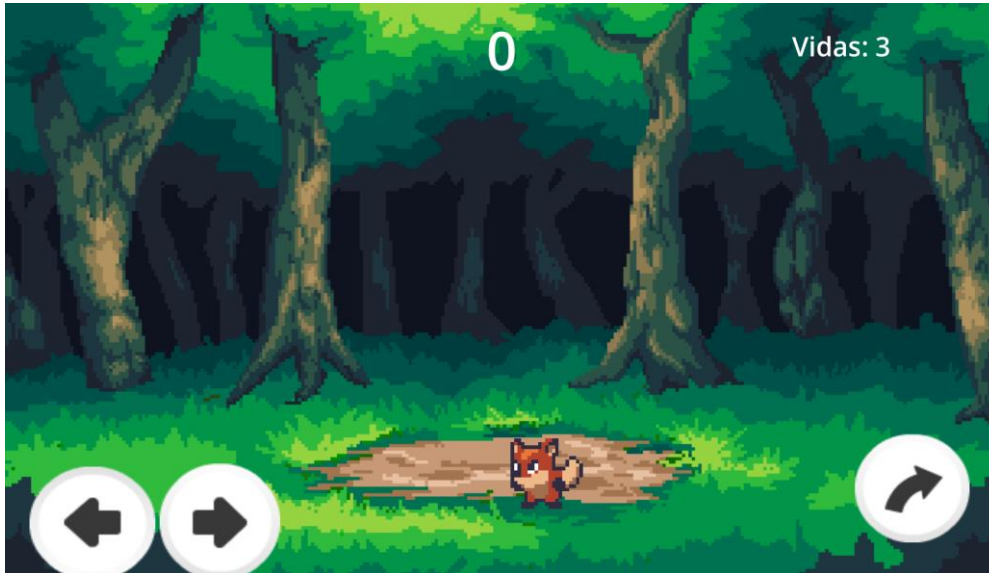
-Muestra opciones para **reiniciar** o **salir al menú**.

```
1  extends CanvasLayer  
2  
3  @onready var boton_reiniciar = $HBoxContainer/Reiniciar  
4  @onready var boton_salir = $HBoxContainer/Salir  
5  
6  
7  func _on_reiniciar_pressed():  
8      >| print("Reiniciando el juego...")  
9      >| get_tree().paused = false  
10     >| get_tree().reload_current_scene() # Reiniciar la escena actual  
11  
12  func _on_salir_pressed():  
13     >| print("Volviendo al menú principal...")  
14     >| get_tree().paused = false  
15     >| get_tree().change_scene_to_file("res://Escenas/menu.tscn") #Salir al menu  
16
```

## 6. CONTROLES MÓVILES.

Configurados en controles.tscn. Se ocultan en PC y aparecen en Android/iOS.

- **Botón izquierda/derecha:** Mueve el personaje.
- **Botón salto:** Permite saltar.
- **En PC** para probarlos tienes que ir dándole clic con el puntero del ratón



## 7. SONIDOS Y ANIMACIONES

- **cogermanzana.mp3** → Se ejecuta en subirScore().
- **morir.mp3** → Se ejecuta en morirse().

```
func subirScore():
    emit_signal("scoreUP")
    if sonido_manzana:
        cogermanzana.play() # Reproducir sonido al coger una manzana
```

```
func morirse():
    vidas -= 1
    actualizar_vidas() # Actualizar el texto de vidas

    if vidas > 0:
        print("▲ Perdiste una vida. Vidas restantes:", vidas)
        posicion_inicial() # Se reinicia el personaje en su posición inicial
    else:
        print("☠ Se acabaron las vidas. Game Over.")
        animated_sprite_2d.play("herido")

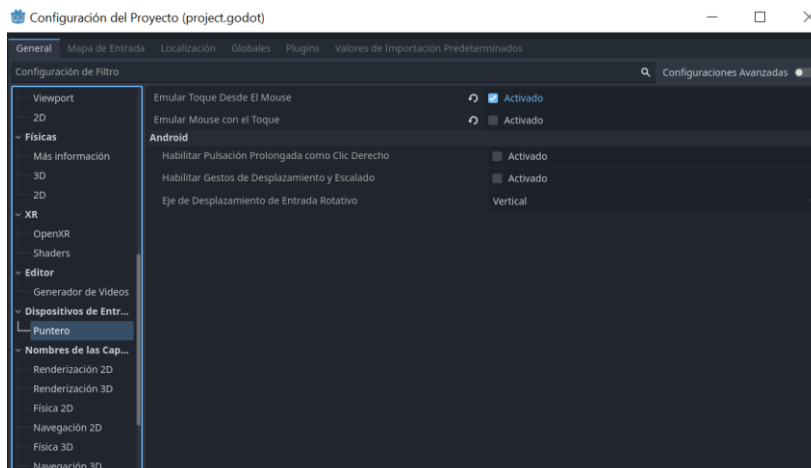
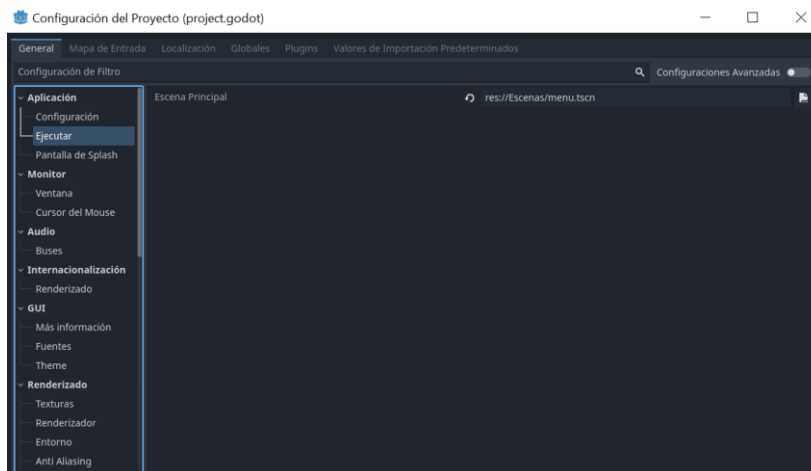
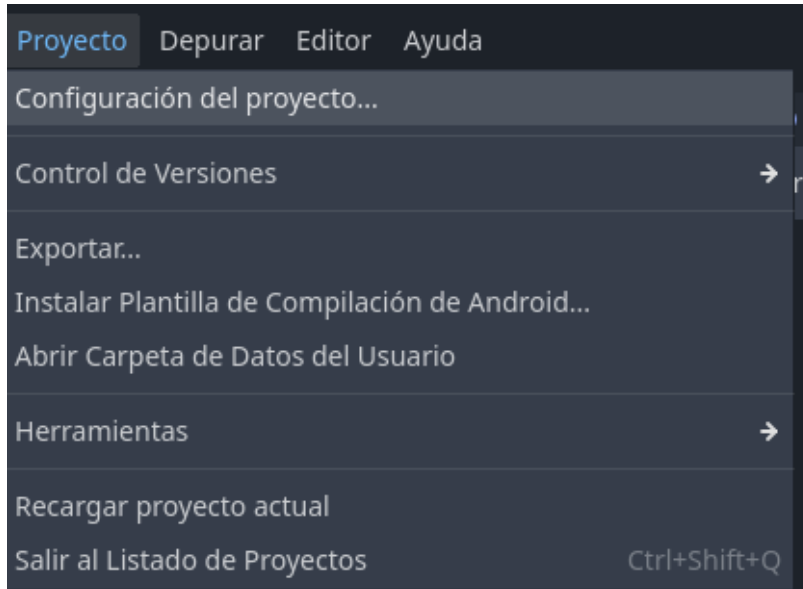
        # 🎵 Reproducir sonido de muerte
        if sonido_morir:
            morir.play()

        emit_signal("playerDied")

        await get_tree().create_timer(0.5).timeout
        get_tree().call_group("UI", "mostrarPantallaMuerte")
```

## 8. CONFIGURACIÓN DEL PROYECTO

- **Menú principal predeterminado:** Configurado en **Proyecto > Configuración > Ejecutar > Escena Principal** (menu.tscn).
- **Entradas del jugador:** Configuradas en **Proyecto > Entrada** (teclado y móvil).



## 9.POSIBLES MEJORAS QUE SE PUEDEN AÑADIR.

- ✓ Agregar **efectos de partículas** al recoger manzanas.
- ✓ Incluir **más enemigos o trampas**.
- ✓ Permitir **skins personalizables** para el personaje.

## 10.ASSETS SACADOS

<https://kenney.nl/assets/onscreen-controls>

[https://es.123rf.com/photo\\_176848630\\_bot%C3%B3n-de-men%C3%BA-de-madera-activo-del-juego-ui-en-estilo-de-dibujos-animados-aislado-sobre-fondo.html](https://es.123rf.com/photo_176848630_bot%C3%B3n-de-men%C3%BA-de-madera-activo-del-juego-ui-en-estilo-de-dibujos-animados-aislado-sobre-fondo.html)

<https://pixabay.com/es/sound-effects/search/recoger/>