

DESARROLLO DE VIDEOJUEGOS CON GODOT ENGINE

Jornada de Integración, Extensión y Actualización
de estudiantes de informática
XI Edición



GODOT



Gastón Caminiti

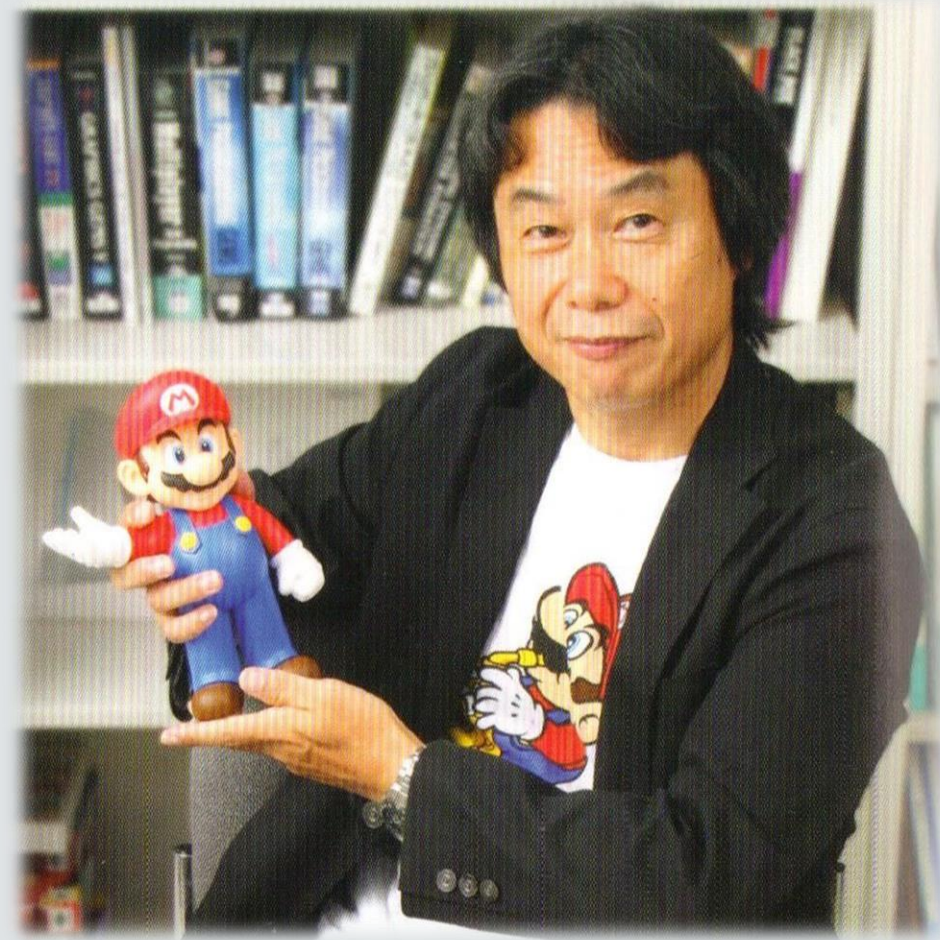
Primera Jornada

- Introducción al Desarrollo de Videojuegos.
- Mecánicas de Videojuegos de Plataforma.
- Introducción a Godot Engine 3.1.
- Nodos y Escenas.
- Creación del avatar.
- Control del personaje.
- Animaciones.

Desarrollo de Videojuegos

*“Yo no me esfuerzo en crear nuevos personajes y nuevos mundos. Sino en crear nuevos **gameplay**”*

Shigeru Miyamoto
(2015)



Desarrollo de Videojuegos - Gameplay

“Gameplay es el resultado de la actividad lúdica producida por un conjunto de desafíos delimitado por reglas y objetivos ”

Nallar Durgan
(2015)

Source: Diseño de juegos en América latina, teoría y práctica I: Estructura lúdica

*Engloba la interacción jugador-juego.
Incluyendo mecánicas, dinámicas y estéticas.*

Desarrollo de Videojuegos - Mecánicas

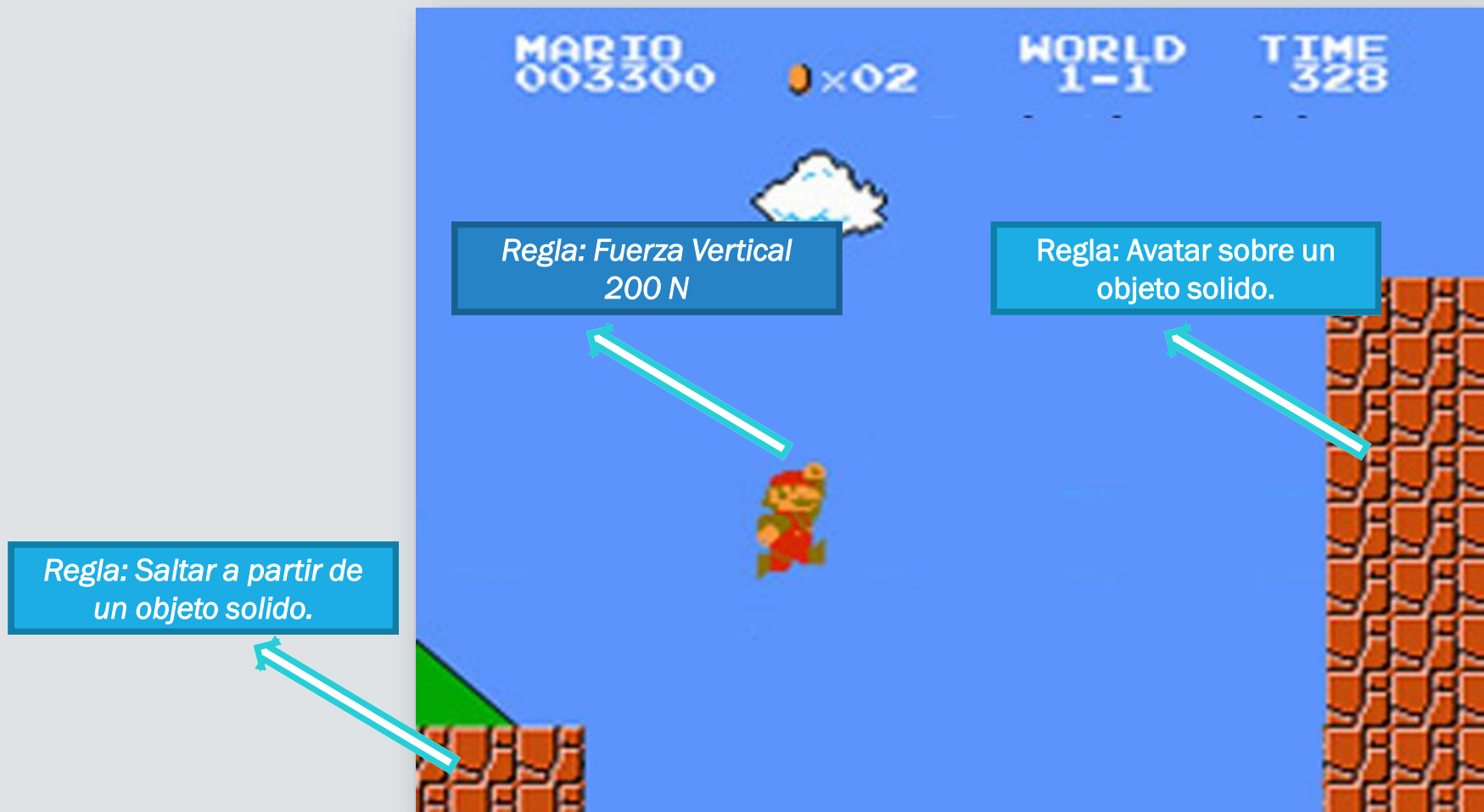
*“Las **mecánicas** son las diferentes acciones, comportamientos y mecanismos de control permitidos al jugador en el juego ”*

Source: MDA: A formal approach to game design and game research

*Lo que podemos hacer en el juego.
Determina la libertad de interacción.*

Mecánicas y reglas

Análisis de la mecánica de Salto en Super Mario Bros.



Mecánicas y reglas

Análisis de la mecánica de Salto en Dark Soul.



Mecánicas y reglas

- Definir las mecánicas de forma única distingue entre acciones y condiciones de juego.
- Las posibilidades de empleo de las mecánicas permiten la aparición y repetición de dinámicas de juego.

Desarrollo de Videojuegos - Dinámicas

“Comportamiento en tiempo de ejecución de las mecánicas controladas por los jugadores”

Source: MDA: A formal approach to game design and game research

Lo que hacemos en el juego, nuestras decisiones y acciones al emplear las mecánicas.

Dinámicas de juego

Análisis de la dinámica de Camping en Counter Strike



Dinámicas de juego

Análisis de la dinámica de Camping en League of Legends

Se busca una situación de superioridad numérica
2vs1 o 3vs2

Se repite la situación
varias veces contra un
enemigo.

El jugador espera a que su
enemigo adopte una mala
posición



Dinámicas de juego

- Situaciones en la interacción jugador-juego.
- Las dinámicas propician experiencias, progresión y aprendizajes al jugador.

Videojuego de Plataformas

Algunas mecánicas núcleo del genero:

- *Correr.*
- *Saltar.*
- *Trepar.*

Videojuego de Plataformas - Ejemplos



Super Meat Boy

Sonic the hedgehog



Enfoque de desarrollo

El motor de videojuegos

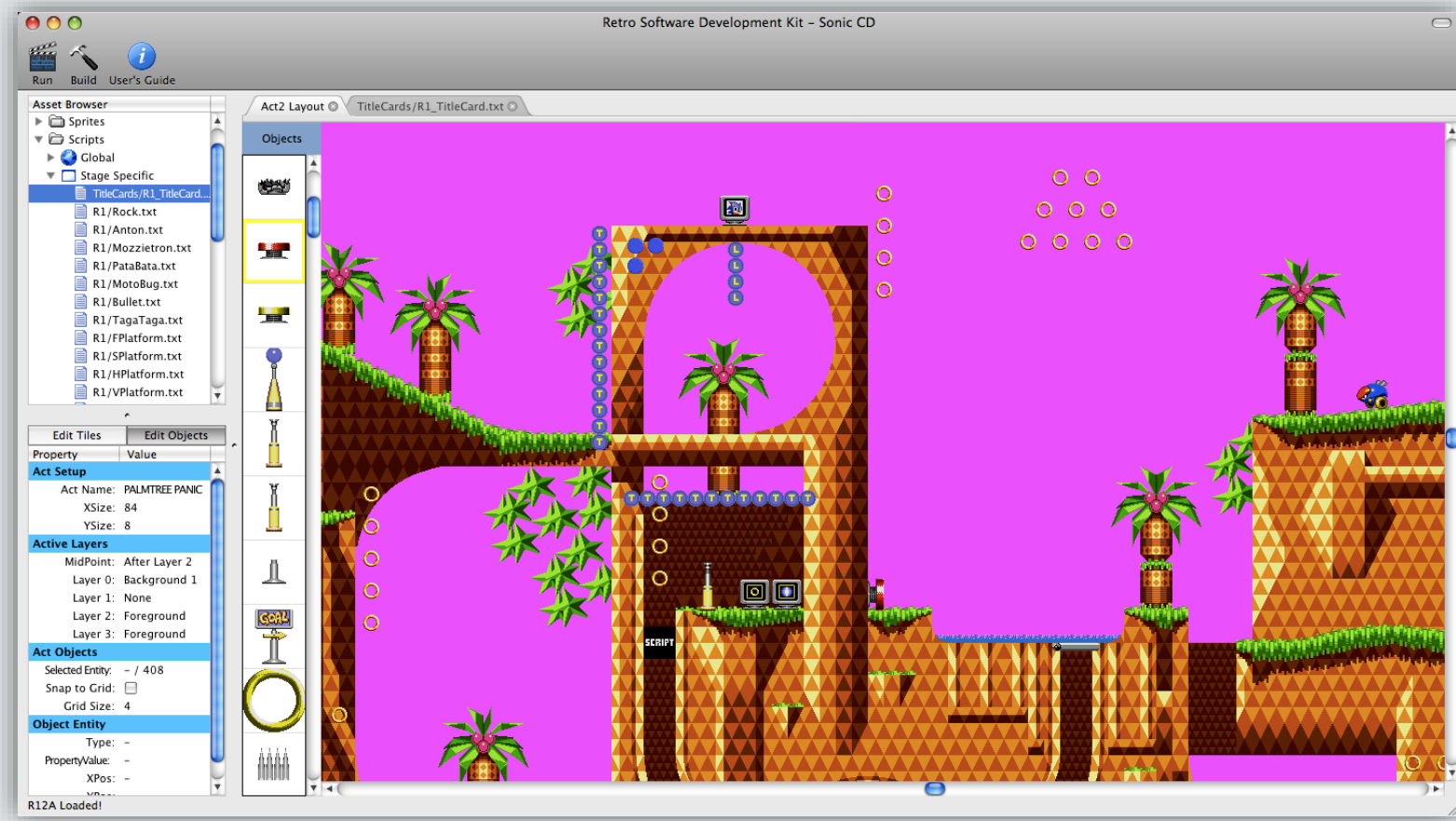
“Solución tecnológica conformada por un conjunto de librerías, sistematizadas e integradas, destinada al desarrollo de videojuegos”

*La elección del Engine debe estar orientada al gameplay.
Escoger la tecnología en sintonía con la meta de diseño.*

Enfoque de desarrollo – Caso Práctico

Retro Engine y Meta de Diseño

“La ausencia de una opción para el desarrollo y porting de juegos de Sonic en consolas modernas impulsa a Christian Whitehead a crear Retro Engine.”

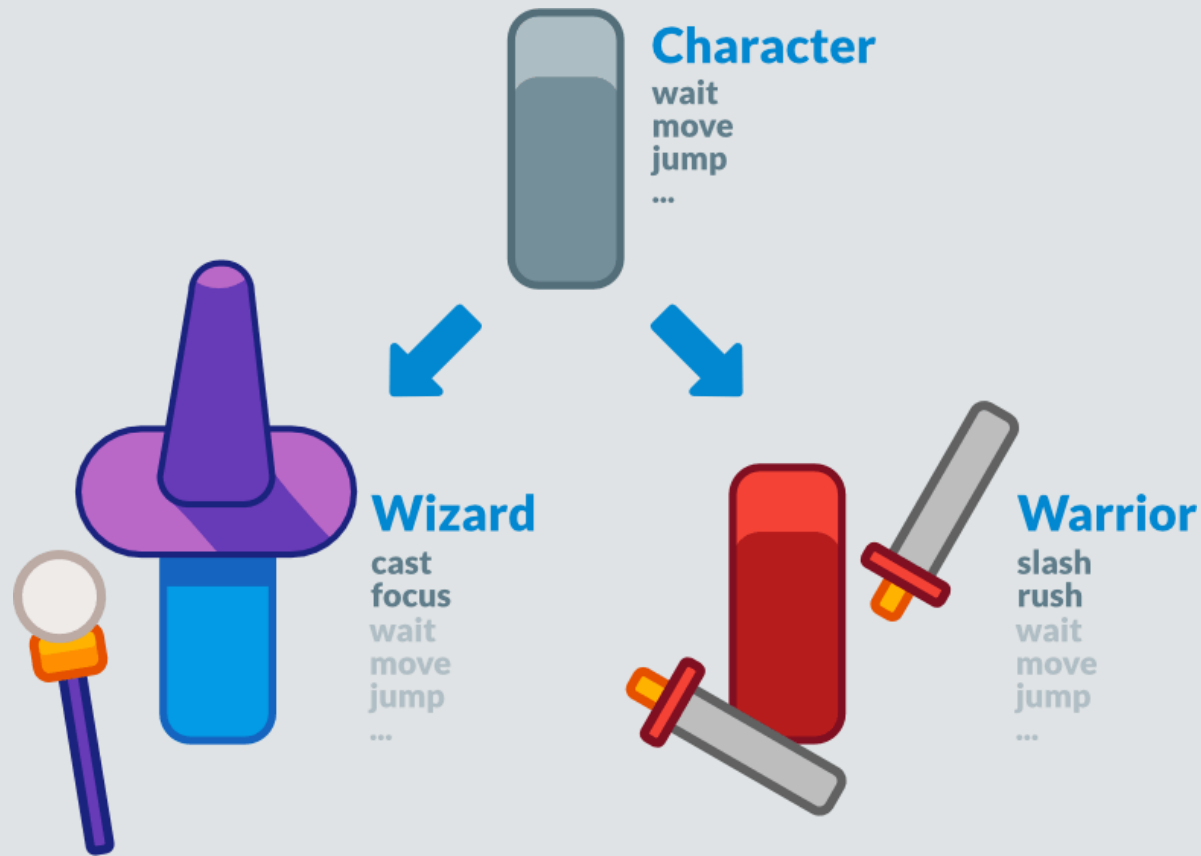


Características de Godot Engine

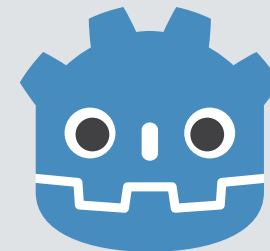
- Software Libre y de Código Abierto (Licencia MIT)
- Multiplataforma: Editor y Publicación de Videojuegos.
- Soporte 2D y 3D.
- Lenguaje de Scripting: GDScript.
- Diseño orientado a objetos.
- Extendible por Plugins y GDNative.



Filosofía de Diseño en Godot Engine



“Godot adopta un diseño orientado a objetos desde el núcleo con su sistema flexible de escenas y jerarquía de nodos.”



GODOT

Source:

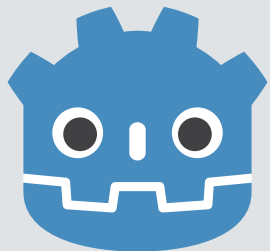
https://docs.godotengine.org/es/latest/getting_started/step_by_step/godot_design_philosophy.html

ACTIVIDADES DEL TALLER

Guía de contenidos para el desarrollo de actividades

Licencia documentación de Godot Engine:

© Copyright 2014-2019, Juan Linietsky, Ariel Manzur and the Godot community (CC-BY 3.0)



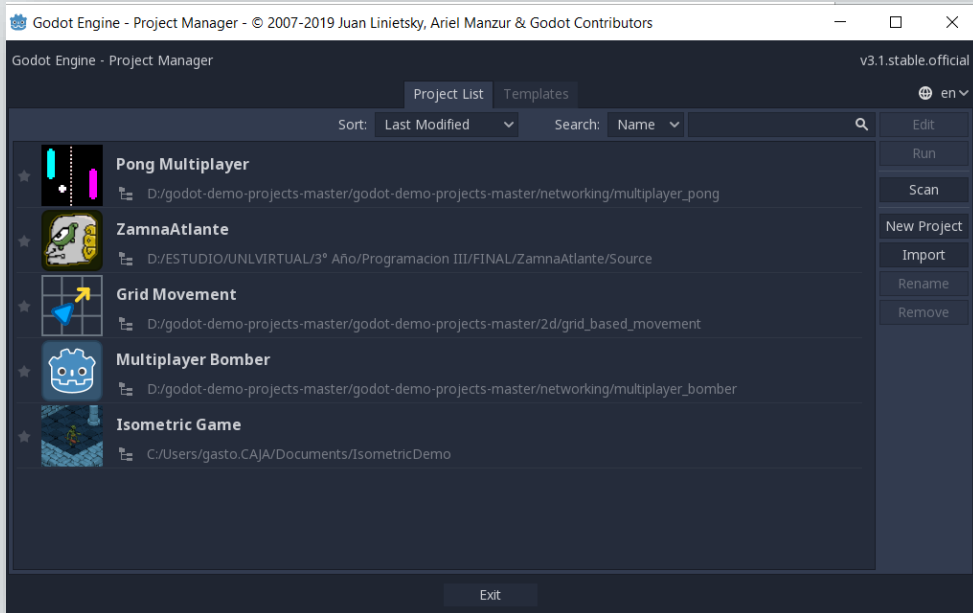
GODOT



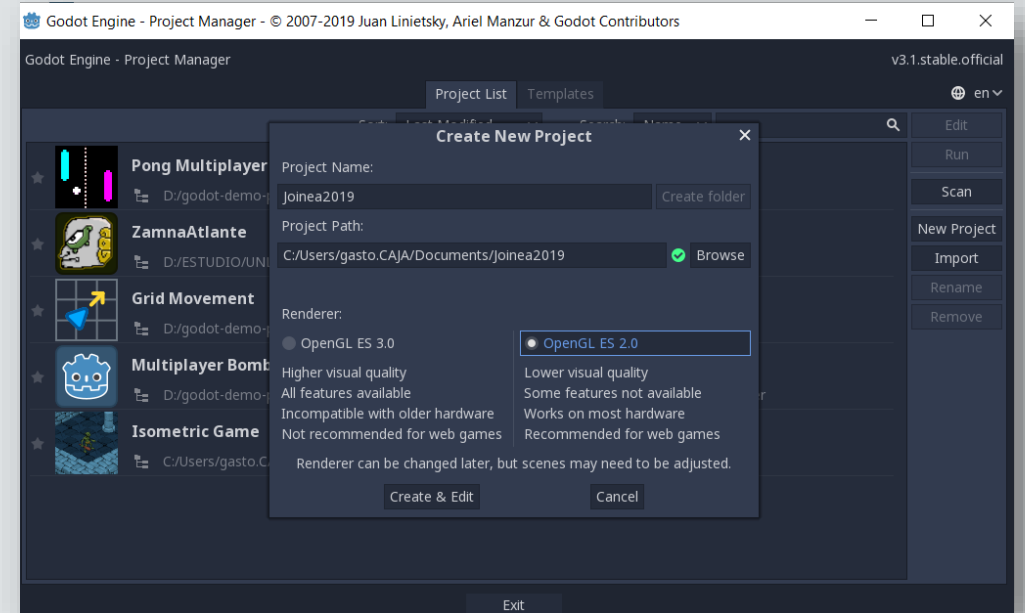
Gastón Caminiti

Creación de Proyecto en Godot Engine

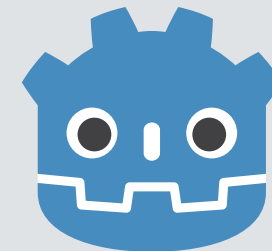
1) Click en “New Project”.



2) Ingresar un nombre de proyecto, click en “Create Folder” y OpenGL ES 2.0.

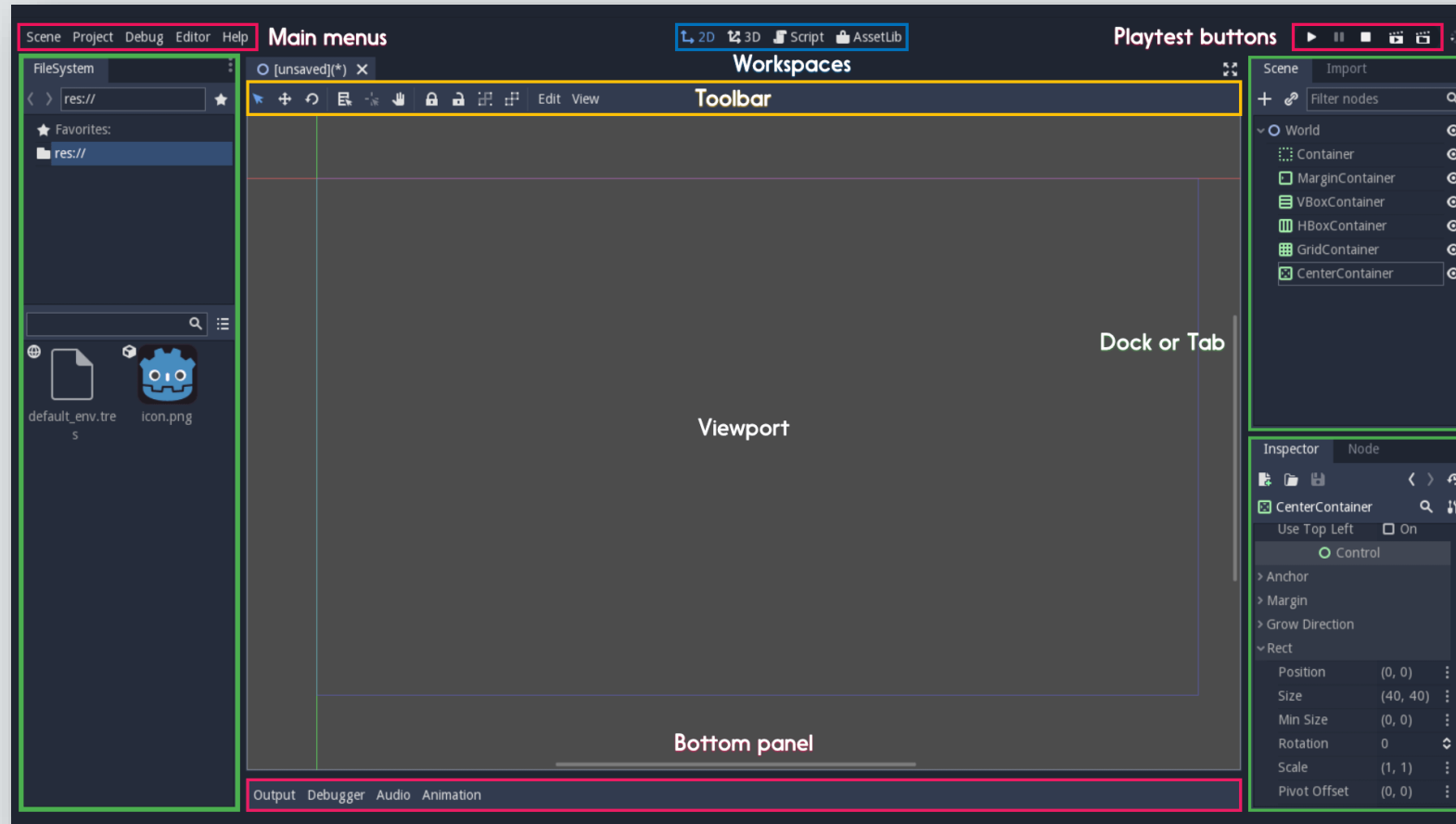


“En la pestaña Templates hay variados ejemplos de proyecto disponibles para descarga”



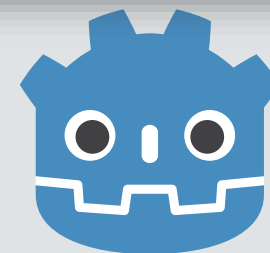
GODOT

Editor de Proyecto en Godot Engine



Source:

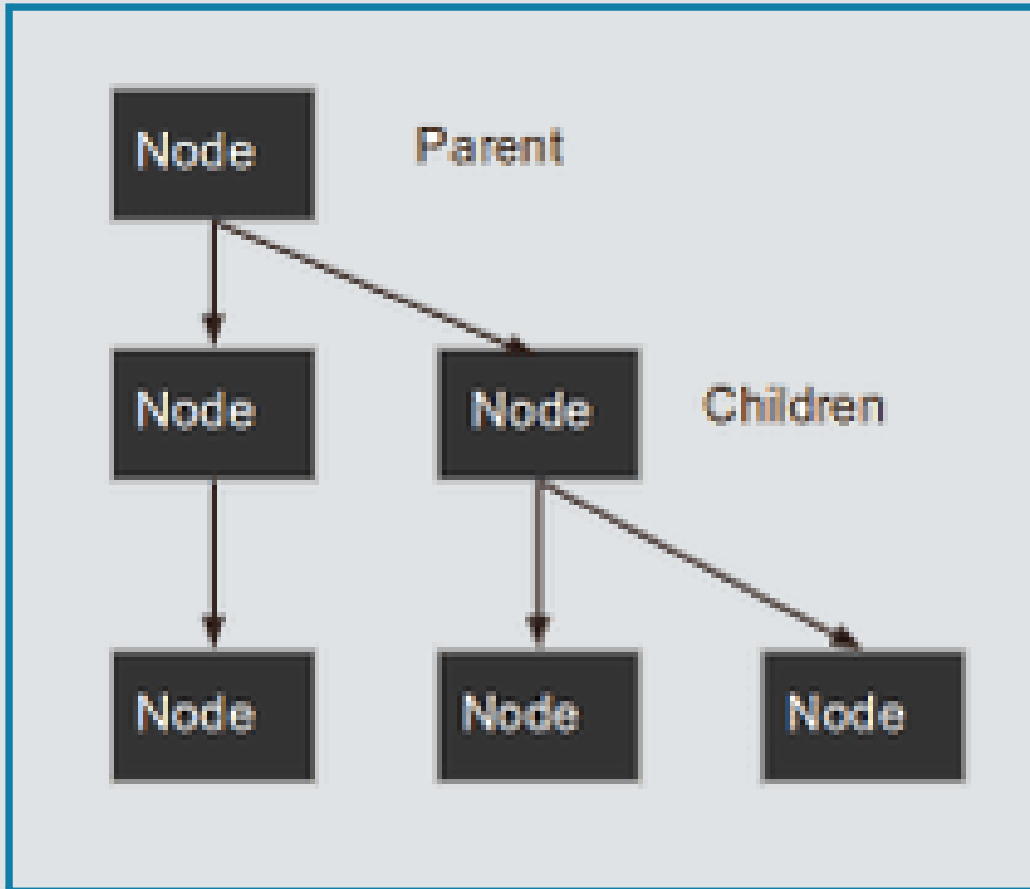
https://docs.godotengine.org/es/latest/getting_started/step_by_step/intro_to_the_editor_interface.html



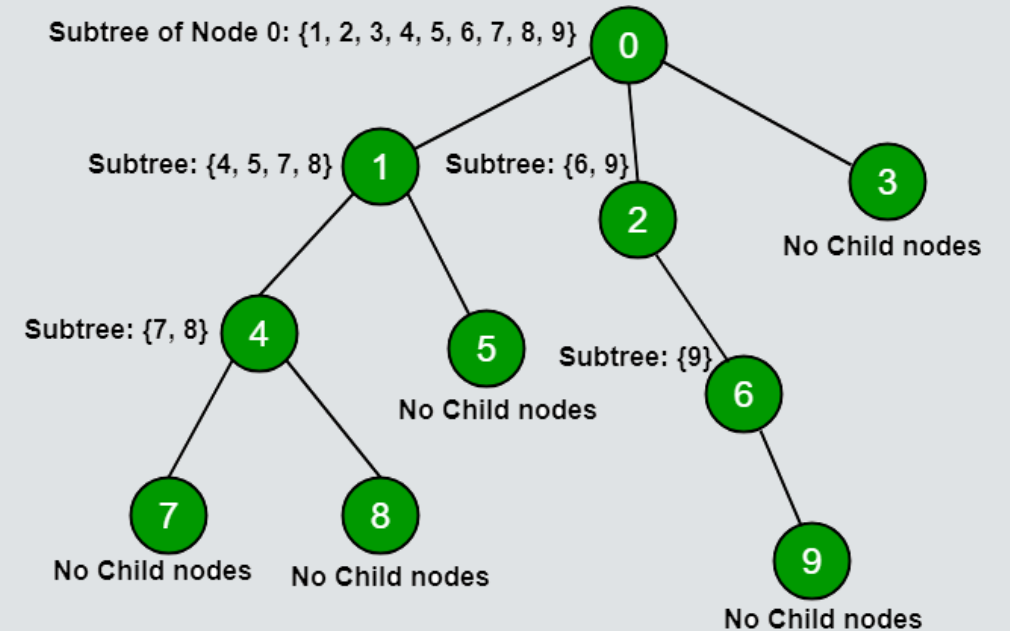
GODOT

Escenas y Nodos en Godot Engine

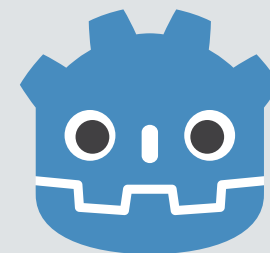
Scene



“Una escena es un árbol jerárquico de nodos ”



Source: https://docs.godotengine.org/es/latest/getting_started/step_by_step/scenes_and_nodes.html

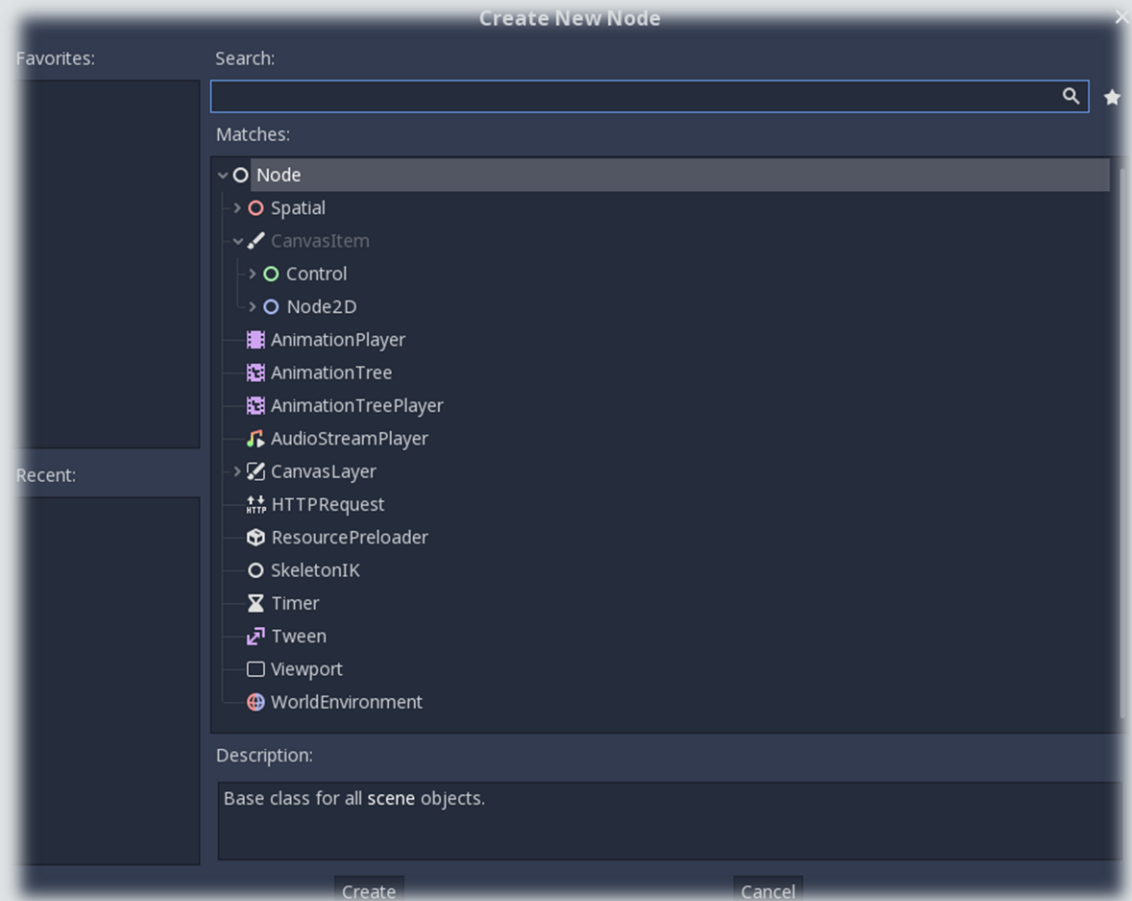


GODOT

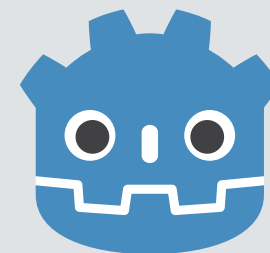
Escenas y Nodos en Godot Engine

“ Los nodos son elementos fundamentales para crear un juego. Desterminan funcionalidades especiales y comportamiento. ”

Combinaremos los distintos tipos de nodos para crear los elementos del juego.



Source: https://docs.godotengine.org/es/latest/getting_started/step_by_step/scenes_and_nodes.html



GODOT

Creación de Objetos – Físicas

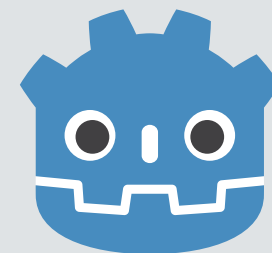
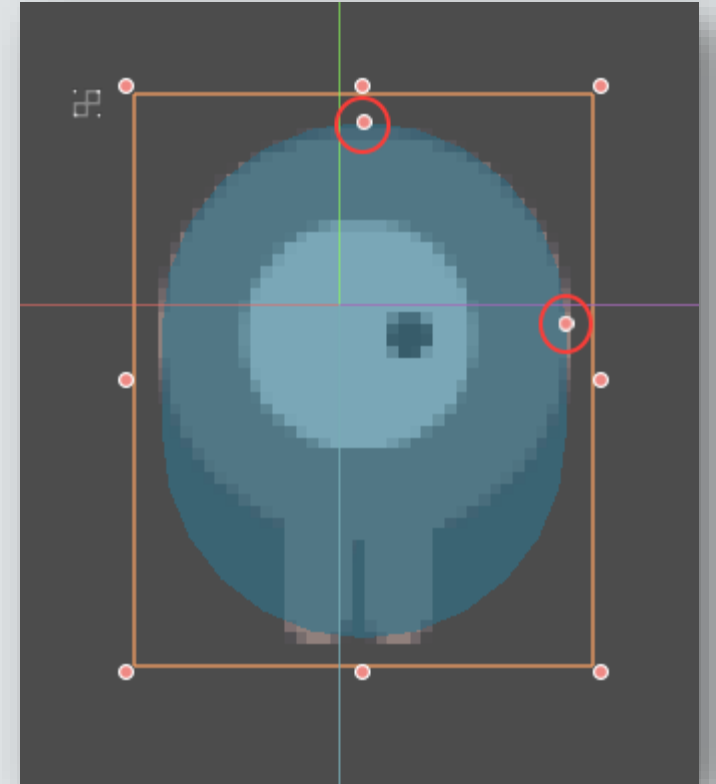
Requerimiento:

Determinar cuando dos objetos del juego entran en contacto. (Colisión).

Aplicación en Godot Engine:

Emplear Nodos que contengan mascaras de colisión.

En función del comportamiento físico de los objetos podemos utilizar nodos: `Area2D`, `StaticBody2D`, `RigidBody2D`, `KinematicBody2D`.



GODOT

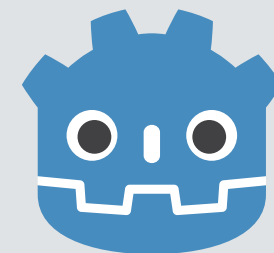
Creación de Objetos Fijos 2D

Requerimiento:

Generar plataformas u obstáculos fijos con los que el avatar colisioné.

Aplicación en Godot Engine:

Emplear el Nodo **StaticBody2D** dado que detectan colisiones pero no se mueve en respuesta a ellas. Además, debemos asignar un **Sprite** representativo a la mascara de colisión (**CollisionShape**).



GODOT

Creación del Avatar – Cuerpo

Requerimiento:

Crear un objeto donde las respuestas a colisiones sean programables.

Aplicación en Godot Engine:

`KinematicBody2D` detecta colisiones con otros cuerpos cuando se mueven, pero no son afectados por propiedades físicas. Hay que codificar el comportamiento requerido.



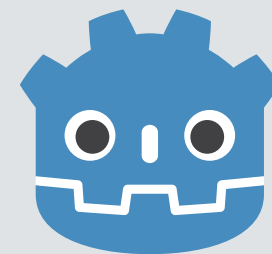
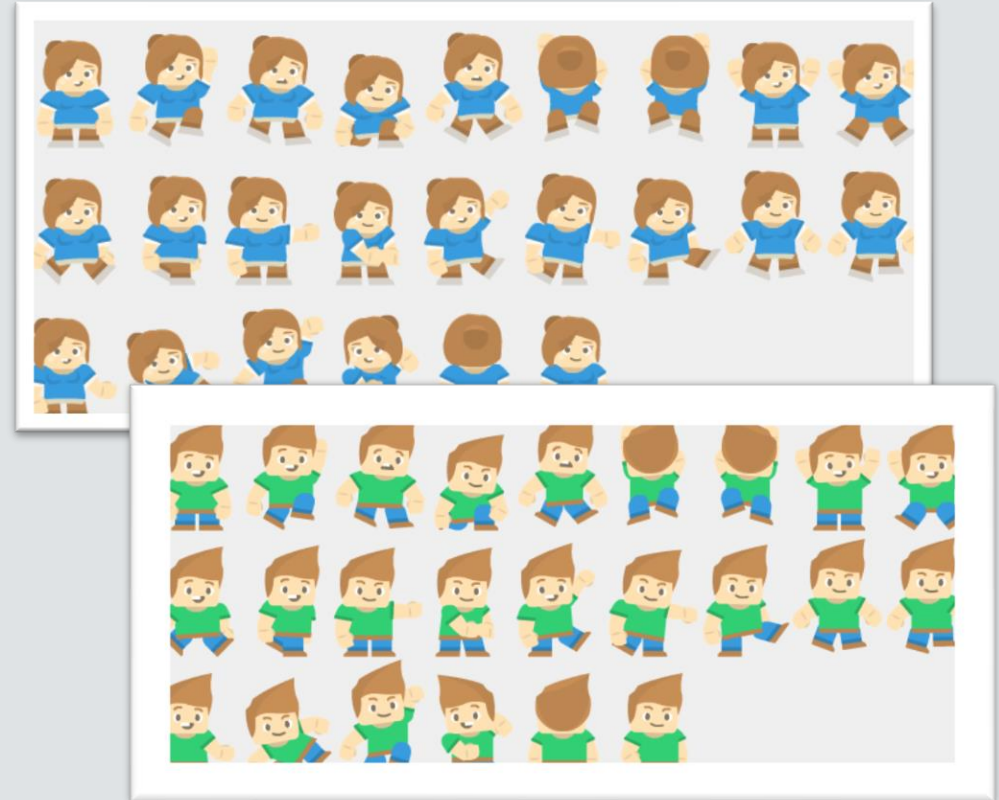
Creación del Avatar – Sprite

Requerimiento:

Asignar textura a elementos distinguibles del videojuego .

Aplicación en Godot Engine:

Emplear el Nodo 2D **Sprite** para gestionar la escala, posición y rotación de la textura. Ya sea de una imagen única o **Spritesheet**.

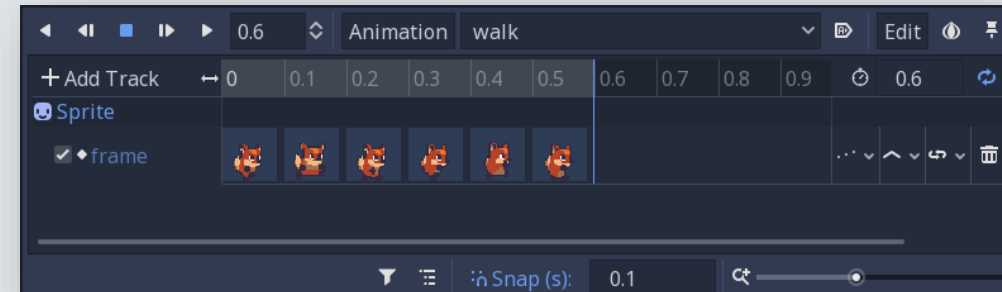


GODOT

Creación del Avatar – Animación

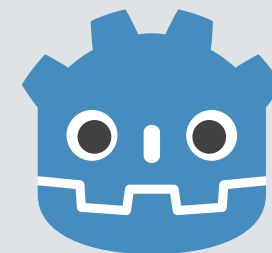
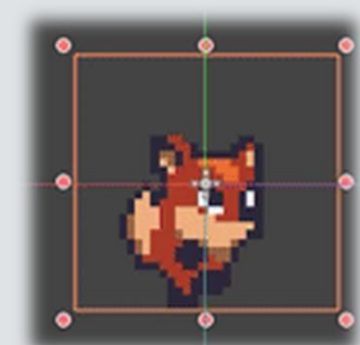
Requerimiento:

Generar animaciones identificables para cada estado del objeto.



Aplicación en Godot Engine:

Utilizar el nodo **AnimationPlayer** en el caso de emplear **Spritesheet** o **AnimatedSprite** si se cuenta con las imágenes que conforman la animación en archivos individuales.



GODOT

Source: https://docs.godotengine.org/es/latest/tutorials/2d/2d_sprite_animation.html

Creación del Avatar – Movimiento 2D

Requerimiento:

Controlar el movimiento del personaje mediante entradas de teclado.

Aplicación en Godot Engine:

Primero, definir las teclas a emplear en la sección Proyecto -> Ajustes del Proyecto.

Asignar un Script al nodo padre y codificar el movimiento del objeto.

Source: https://docs.godotengine.org/es/latest/tutorials/2d/2d_movement.html

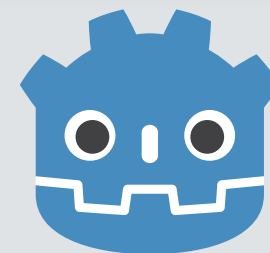
```
extends KinematicBody2D

export (int) var speed = 200

var velocity = Vector2()

func get_input():
    velocity = Vector2()
    if Input.is_action_pressed('right'):
        velocity.x += 1
    if Input.is_action_pressed('left'):
        velocity.x -= 1
    if Input.is_action_pressed('down'):
        velocity.y += 1
    if Input.is_action_pressed('up'):
        velocity.y -= 1
    velocity = velocity.normalized() * speed

func _physics_process(delta):
    get_input()
    velocity = move_and_slide(velocity)
```



GODOT

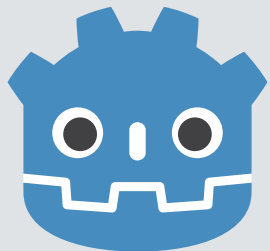
¡GRACIAS POR PARTICIPAR!

Recursos del Taller:

<https://github.com/gastoncaminiti/joinea2019>

Licencia documentación de Godot Engine:

© Copyright 2014-2019, Juan Linietsky, Ariel Manzur and the Godot community (CC-BY 3.0)



GODOT



Gastón Caminiti