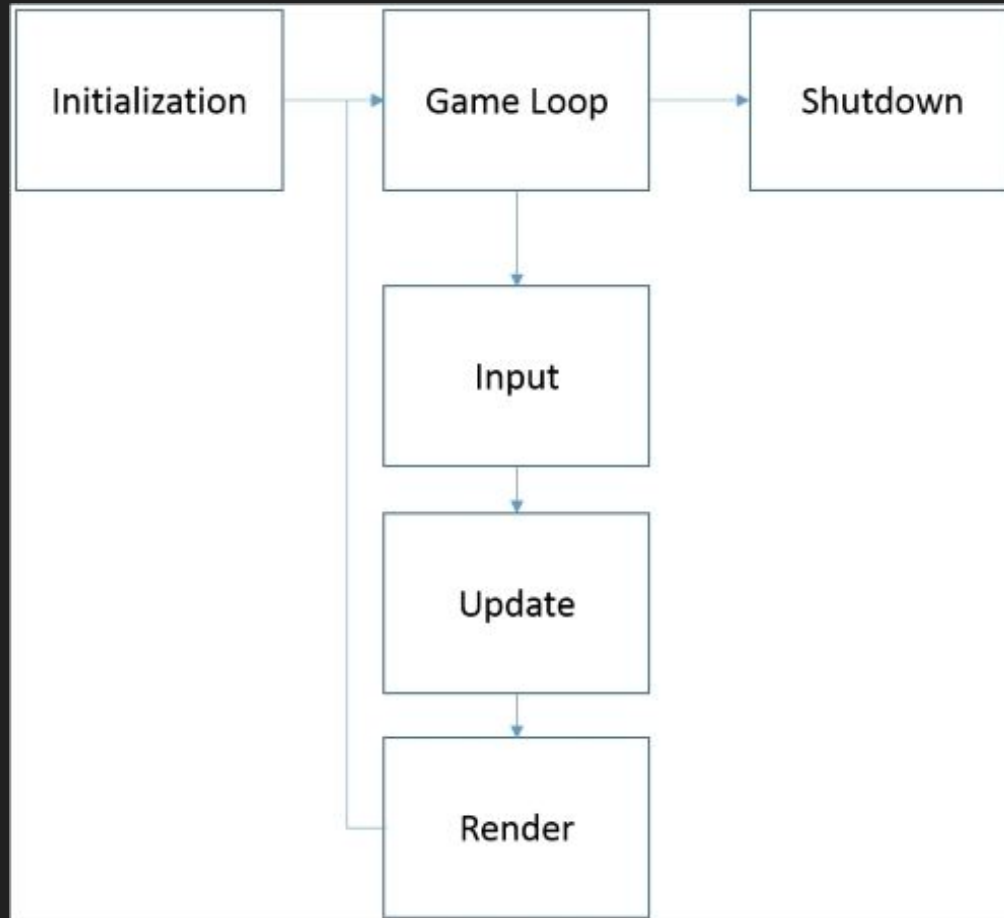


Motores de juegos y Godot Engine

Interfícies Gràfiques d'Usuari

Concepto de Game Loop





GODOT

Game Engine

Objectives

Devices: 3 / 3

Enemies: 14 / 15

Godot Engine

Godot Engine es un motor de videojuegos multiplataforma con múltiples características para crear juegos 2D y 3D desde una interfaz unificada. Él provee un conjunto exhaustivo de herramientas comunes para que los usuarios puedan enfocarse en crear juegos sin tener que reinventar la rueda. Juegos que pueden exportarse en un sólo clic a numerosas plataformas, incluyendo las principales plataformas de escritorio (Linux, macOS, Windows), móviles (Android, iOS) y basadas en la web (HTML5).

Godot es completamente gratuito y de código abierto bajo la licencia permisiva del MIT. Sin condiciones, sin regalías, nada. Los juegos de los usuarios son suyos, hasta la última línea del código del motor. El desarrollo de Godot es totalmente independiente y dirigido por la comunidad, lo que permite a los usuarios ayudar a dar forma a su motor para que coincida con sus expectativas. Está respaldado por Software Freedom Conservancy sin fines de lucro.

Història

<https://godotengine.org/article/retrospective-and-future>

La iniciativa començó en el any 2010 a mans de Juan Linietsky y Ariel Manzur. Trabajaron con Okam desarrollando videojuegos.

- Godot 1.0: 2014
- Godot 2.0: 2016
- Godot 3.0: 2018
- Versión estable actual: Godot 3.4.3
- Versión más reciente: Godot 4.0 alpha 5

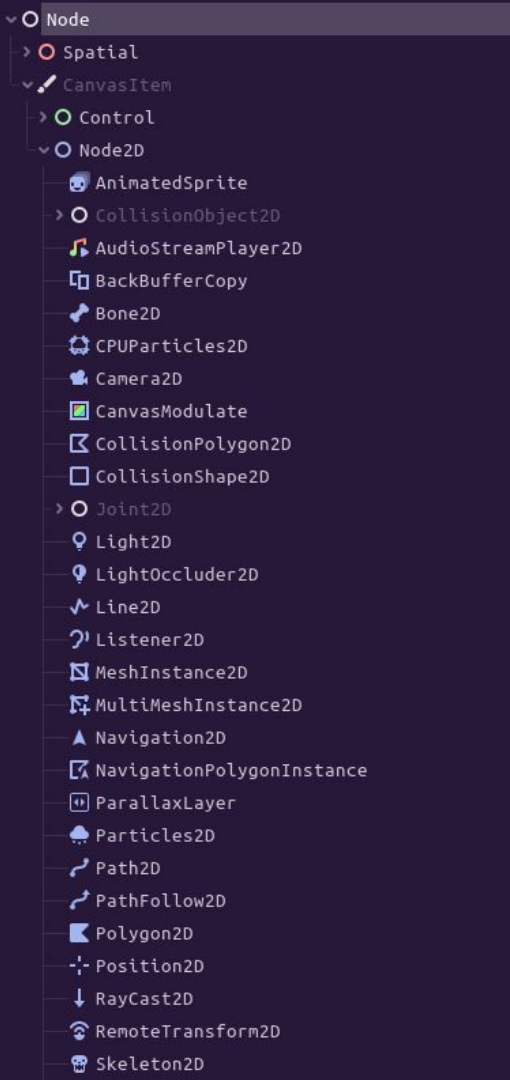
Nodos

Son el bloque de construcción fundamental de tu juego.

Tienen las propiedades básicas necesarias para implementar algún tipo de funcionalidad.

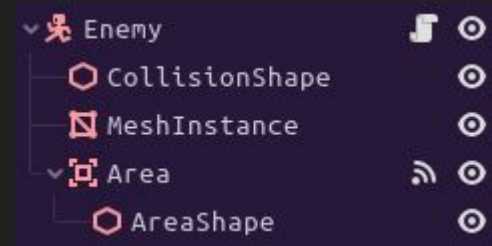
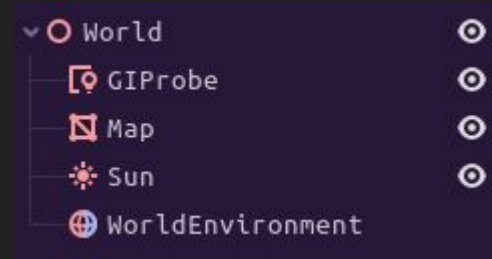
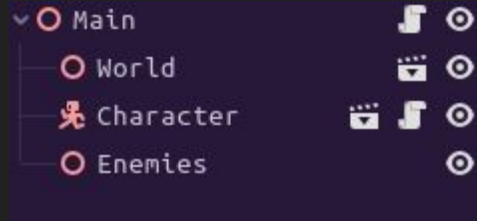
Tienen:

- Un nombre.
- Propiedades editables.
- Reciben callbacks para actualizar cada fotograma.
- Puedes extenderlos con nuevas propiedades y funciones.
- Puedes agregarlos a otro nodo como un hijo.



Escena

Organización de nodos en forma de árbol.



Lenguajes

Godot ofrece **cinco lenguajes de programación de juegos** a través de su tecnología GDNative:

- **GScript**
- **C#**
- **VisualScript**
- **C++**
- **C**

Manejados por la comunidad:

- **D**
- **Kotlin**
- **Num**
- **Python**
- **Rust**

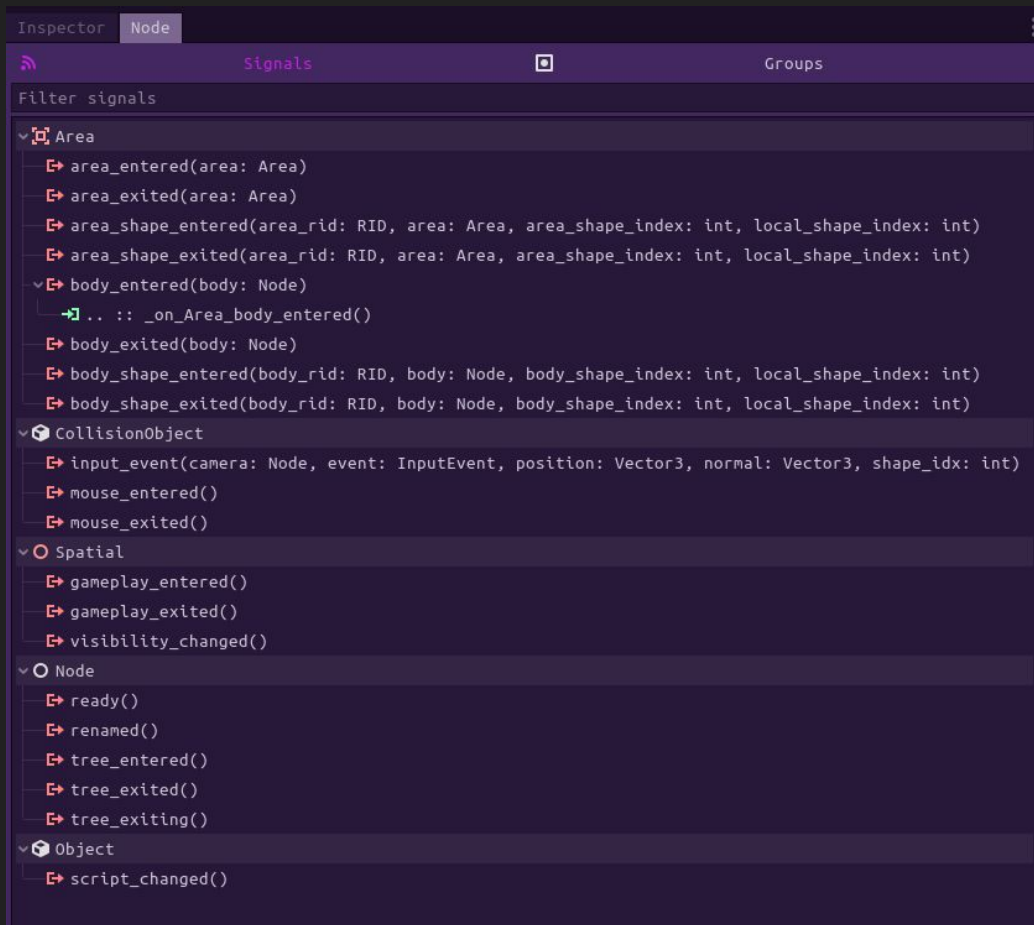
GScript

- Orientado a objetos.
- Imperativo.
- Una sintaxis simple que conduce a archivos ligeros.
- Tiempos de carga y compilación ultrarrápidos.
- Estrecha integración del editor, con finalización de código para nodos, señales y más información de la escena a la que está conectado.
- Tipos de transformación y vector incorporados, lo que lo hace eficiente para el uso intensivo de álgebra lineal, imprescindible para los juegos.
- Admite múltiples hilos con la misma eficacia que los lenguajes escritos estáticamente. Esta es una de las características que no podríamos proporcionar fácilmente con un lenguaje de terceros como Lua o Python.
- Sin Garbage Collector. La gestión de memoria la realiza el programador.
- Escritura gradual. Permite la inferencia de tipos o un tipado fuerte.

Señales

Son mensajes que emiten los nodos cuando les sucede algo específico. Otros nodos pueden conectarse a esa señal y llamar a una función cuando ocurra el evento.

Es un mecanismo de delegación integrado en Godot que permite que un objeto del juego reaccione a un cambio en otro sin que se refieran entre sí. El uso de señales limita el acoplamiento y mantiene su código flexible.



Recursos interesantes

Descarga:

- <https://godotengine.org>

Referencias a la documentación:

- <https://docs.godotengine.org/en/stable/>
- <https://docs.godotengine.org/en/stable/tutorials/networking/websocket.html>

¿Qué puede hacerse con Godot Engine?

- <https://godotengine.org/showcase>

Anem per feina

Repositorio de GitHub:

<https://github.com/EnergyWox/GodotShowcase>