



ANTEPROYECTO

Creación de juegos con Godot 4.4

**Ciclos Formativo de Grado Superior de
Desarrollo de Aplicaciones
Multiplataforma**

Nombre:	Hugo del Rey Holgueras
e-mail:	hugo.reyhol@educa.jcyl.es
Código Curso:	IFC02S

Tabla de contenido

1. Motivación del proyecto.....	3
1.1 Situación inicial:.....	3
A. Motores de videojuegos.....	3
B. Motores en el mercado.....	3
C. ¿Por qué Godot?.....	4
D. ¿Qué es un GDD?.....	5
E. Proceso de creación de un videojuego.....	5
2. Objetivos.....	7
3. Plan de trabajo.....	7
4. Medios a usar.....	8
5. Tiempo Estimado para su realización.....	8
6. Bibliografía.....	9

1. Motivación del proyecto.

He elegido este proyecto porque siempre me han gustado los videojuegos y me gustaría aprender a hacerlos, además Godot es un motor emergente en el cual veo mucho potencial de que en un futuro podrá hacerle frente a motores líderes de la industria como Unity y Unreal, especialmente en los juegos enfocados al 2D como el que voy a realizar.

1.1 Situación inicial:

A. Motores de videojuegos

Un motor de videojuegos es un software diseñado para facilitar el desarrollo de videojuegos, proporcionando un conjunto de herramientas y funcionalidades que simplifican tareas como la renderización de gráficos, la gestión de físicas, la detección de colisiones y la integración de sonido. En lugar de programar desde cero cada aspecto del juego, los desarrolladores pueden utilizar estos motores para ahorrar tiempo y recursos, permitiéndoles enfocarse en la jugabilidad y el diseño.

El uso de motores de videojuegos es fundamental en la industria, ya que optimizan el desarrollo, permiten la reutilización de código y facilitan la compatibilidad con múltiples plataformas.

B. Motores en el mercado

Existen diversos motores de videojuegos en el mercado, cada uno con sus propias características, ventajas y casos de uso. Entre los más populares destacan **Unity**, **Unreal Engine** y **Godot**, que han sido utilizados en la creación de numerosos títulos exitosos en la industria.

- **Unity:** Es uno de los motores más utilizados a nivel mundial, gracias a su versatilidad y facilidad de uso. Soporta múltiples plataformas, desde PC y consolas hasta dispositivos móviles y realidad virtual. Su lenguaje principal es C#, y ofrece una gran cantidad de recursos y documentación, lo que lo hace ideal tanto para principiantes como para desarrolladores avanzados. Es ampliamente utilizado en juegos 2D y 3D.
- **Unreal Engine:** Es conocido por su impresionante capacidad gráfica y su motor de físicas avanzado. Utiliza Blueprints, un sistema de programación visual, además de admitir C++ para un mayor control sobre el desarrollo. Es el motor elegido para juegos de alto rendimiento y realismo gráfico, siendo ideal para proyectos 3D de gran escala, aunque también permite desarrollar juegos 2D.
- **Godot:** Godot es un motor de código abierto que ha ganado popularidad en los últimos años. Soporta tanto juegos 2D como 3D y utiliza su propio lenguaje de scripting, GDScript, similar a Python, aunque también admite

C# y C++. Su arquitectura basada en nodos y su ligereza lo convierten en una opción excelente para desarrolladores independientes y juegos 2D.

C. ¿Por qué Godot?

Godot es una excelente elección debido a su enfoque en juegos 2D, su ligereza y su facilidad de uso. A diferencia de otros motores, como Unity o Unreal Engine, Godot está diseñado con una arquitectura que optimiza el rendimiento en juegos bidimensionales, lo que lo hace ideal para proyectos como este.

Ventajas de usar Godot:

1. Optimización para juegos 2D

Aunque muchos motores soportan desarrollo en 2D, Godot cuenta con un motor exclusivo para este tipo de juegos, lo que permite un mejor rendimiento y una mayor facilidad en la implementación de físicas, colisiones y animaciones.

2. Código abierto y gratuito

A diferencia de otros motores, Godot es completamente de código abierto y no requiere pagos por licencias ni regalías. Esto lo convierte en una opción accesible para desarrolladores independientes y principiantes.

3. Lenguaje de programación sencillo y flexible

Godot utiliza **GScript**, un lenguaje similar a Python, que es fácil de aprender y permite una programación rápida y eficiente. Este nos permite probar el juego rápidamente en el editor ya que es un lenguaje interpretado, además al exportar un juego el código GScript se compila a **bytecode**, un formato binario optimizado que puede ser ejecutado rápidamente por la máquina virtual de Godot, mejorando el rendimiento y protegiendo el código fuente del producto final.

4. Interfaz intuitiva y arquitectura basada en nodos

La estructura de Godot se basa en un sistema de nodos, lo que facilita la organización del proyecto y simplifica la creación de escenas y elementos del juego.

5. Multiplataforma y ligero

Godot permite exportar juegos a Windows, macOS, Linux, Android, iOS y HTML5 sin complicaciones. Además, su tamaño es considerablemente menor en comparación con otros motores, lo que permite un desarrollo ágil sin consumir demasiados recursos.

D. ¿Qué es un GDD?

El **Game Design Document (GDD)** es un documento que recopila toda la información esencial sobre el diseño y desarrollo de un videojuego. Su propósito es servir como una guía estructurada para el equipo de desarrollo, asegurando que todos los aspectos del juego, desde su mecánica hasta su estética, estén claramente definidos antes y durante la producción.

Un **GDD** puede variar en complejidad dependiendo del tamaño y alcance del juego, pero generalmente incluye los siguientes elementos:

1. Concepto y visión

- Resumen del juego: género, historia, ambientación y objetivos.
- Público objetivo y plataformas en las que se lanzará.

2. Mecánicas y jugabilidad

- Controles, reglas y progresión del juego.
- Física y comportamiento de los personajes o elementos interactivos.
- Diseño de niveles y estructura del juego.

3. Personajes y elementos gráficos

- Descripción de los personajes principales y enemigos.
- Estilo artístico y referencias visuales.

4. Sonido y música

- Efectos de sonido y música ambiental.
- Voz y diálogos en caso de haberlos.

5. Interfaz de usuario (UI/UX)

- Menús, pantallas de inicio, configuraciones, etc...

6. Tecnología y herramientas

- Motor de juego utilizado.
- Lenguaje de programación y requisitos técnicos.

E. Proceso de creación de un videojuego

El desarrollo de un videojuego es un proceso complejo que involucra varias las siguientes etapas:

1. Preproducción

Esta etapa es clave para definir la visión del juego y planificar su desarrollo.

- **Idea y concepto:** Se define la premisa del juego, su género, mecánicas principales y público objetivo.
- **Game Design Document (GDD):** Se crea el documento de diseño del juego, que servirá como guía para el equipo.
- **Selección de herramientas:** Se elige el motor de juego, los lenguajes de programación y las herramientas de diseño gráfico y sonido.

2. Prototipado

En esta fase se desarrolla un prototipo básico para probar las mecánicas principales del juego.

- **Creación de una Versión Mínima Jugable (MVP):** Se implementan las funciones esenciales, como el control del personaje y la detección de colisiones.
- **Pruebas y ajustes:** Se identifican posibles mejoras en la jugabilidad y se ajustan las mecánicas antes de avanzar a la producción completa.

3. Producción

Esta es la etapa más extensa y en la que el juego toma forma.

- **Programación:** Se implementa el código para la lógica del juego, física, IA y demás funcionalidades.
- **Diseño de niveles:** Se crean los escenarios y se ajusta la dificultad.
- **Arte y sonido:** Se diseñan los gráficos, animaciones, efectos visuales y sonido.
- **Optimización:** Se mejora el rendimiento del juego para asegurar que funcione sin problemas en distintas plataformas.

4. Pruebas y Corrección de Errores

Antes de lanzar el juego, se realizan pruebas para detectar y corregir errores.

- **Pruebas internas:** Se verifica el correcto funcionamiento de todas las mecánicas.
- **Feedback de jugadores:** Se realizan pruebas con usuarios para obtener opiniones sobre la jugabilidad y hacer ajustes finales.

5. Publicación y Distribución

Una vez finalizado el desarrollo, el juego se exporta y se distribuye en las plataformas elegidas.

- **Exportación:** En Godot, el juego se compila y empaqueta para su distribución en PC, móviles o web.

- **Marketing y lanzamiento:** Se promociona el juego y se publica en tiendas digitales como Steam, Itch.io o la App Store.

6. Mantenimiento y Actualizaciones

Después del lanzamiento, es común continuar con el mantenimiento del juego.

- **Corrección de bugs:** Se solucionan errores que puedan surgir tras el lanzamiento.
- **Actualizaciones de contenido:** Se añaden mejoras, nuevos niveles o funcionalidades según el feedback de los jugadores.

2. Objetivos.

A nivel personal quiero aprender a realizar videojuegos ya que es un hobby que tengo y a nivel profesional me ayudará a organizar proyectos complejos en el futuro.

3. Plan de trabajo.

1. Definición de la idea

En este apartado se definirá todo lo que queremos realizar en el proyecto.

a. Concepto básico

La idea del videojuego, en este caso haré un **clon de Flappy Bird** para poder centrarme en la implementación de las distintas tecnologías que voy a usar.

b. Mecánicas principales

Las mecánicas serán las más parecidas posibles a la del juego original, la pantalla avanzará automáticamente hacia la derecha en la cual aparecerán obstáculos que el jugador deberá evitar para seguir avanzando, para hacer esto el jugador realizará una acción que le permitirá elevarse y el personaje descenderá gradualmente por el efecto de la gravedad.

Si el jugador choca con un obstáculo el nivel se reiniciará, se guardará en una base de datos las distintas puntuaciones de los usuarios para hacer un ranking que motive a los jugadores a intentar llegar más lejos.

c. Estilo visual

Se usarán los gráficos del juego original.

d. Plataformas de lanzamiento

El proyecto se exportará a Windows, Linux, Android y Web.

2. Realización de un GDD (Game Desing Document) sencillo

Se realizará un documento explicando a mayor profundidad todo lo comentado en el apartado 1. *Definición de la idea*.

3. Investigación sobre el funcionamiento de Godot

Para poder realizar el juego de forma eficiente se investigará como implementar las distintas funcionalidades que incluya en el GDD.

4. Creación de un prototipo

Se creará un prototipo de la idea para comprobar su funcionalidad.

5. Pruebas y mejoras de las mecánicas

Se probarán todas las mecánicas implementadas y se realizarán los ajustes necesarios a estas para que el juego sea divertido de jugar.

6. Creación del juego a partir del prototipo

Una vez que este conforme con el prototipo se implementarán los detalles que mejoren la experiencia de usuario como una interfaz intuitiva, animaciones, sonidos, etc...

7. Corrección de errores

Se solucionarán todos los errores que encuentre en el prototipo.

8. Compilación y exportación del juego

Una vez el proyecto esté finalizado se compilará y exportara para que se pueda ejecutar en distintas plataformas sin necesidad de instalar Godot en ellas.

4. Medios a usar.

Hardware:

- **Equipo de desarrollo:** Mi ordenador personal.
- **Servidor:** El servidor correrá en local.

Software:

- **Godot 4.4:** Para la programación del juego.
- **Aseprite v1.3.10.1:** Para modificar y crear imágenes pixel-art para los visuales del juego.
- **Audacity 3.4.2:** Para modificaciones en archivos de audio.

5. Tiempo Estimado para su realización.

Estimo que la duración del proyecto serán aproximadamente unas 50 horas.

6. Bibliografía.

[Documentacion Godot](#)