

Diseño y desarrollo de un RPG con Unity

Autor: Bernat Rodríguez Caldú

Tutor: Manel Fernández Rodríguez

Profesor: Joan Arnedo Moreno

Grado de Multimedia

Estudios de Informática, Multimedia y Telecomunicación

18 de junio de 2023

Créditos/Copyright

  
Esta obra está sujeta a una licencia de Reconocimiento - NoComercial - SinObraDerivada

[3.0 España de CreativeCommons.](http://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

|  |  |
| --- | --- |
| **Título del trabajo:** | *Diseño y desarrollo de un RPG con Unity* |
| **Nombre del autor:** | *Bernat Rodríguez Caldú* |
| **Nombre del colaborador/a docente:** | *Manel Fernández Rodríguez* |
| **Nombre del PRA:** | *Joan Arnedo Moreno* |
| **Fecha de entrega (mm/aaaa):** | *06/2023* |
| **Titulación o programa:** | *Grado de Multimedia* |
| **Área del Trabajo Final:** | *Videojuegos* |
| **Idioma del trabajo:** | *Español* |
| **Palabras clave** | *videojuego, RPG, unity* |
| **Resumen del Trabajo:** | |
| La industria del videojuego es un sector económico que se encuentra en pleno apogeo y en el que cada vez es más sencillo iniciarse. En Internet encontramos numerosos recursos, entre ellos: plataformas, tutoriales, assets, iconos, texturas, animaciones, efectos de audio, etc.  En este proyecto se presenta el **proceso de diseño y desarrollo de un** videojuego de rol o **RPG** (*role-playing game*) mediante el uso de Unity, uno de los motores de videojuegos más utilizados hoy en día por principiantes debido a su óptima curva de aprendizaje y versatilidad. Asimismo, nos familiarizamos con el lenguaje de programación C# y hacemos uso de otros programas en las diferentes etapas del diseño tales como GIMP en la creación de texturas o Audacity en la producción de efectos de sonido. También utilizamos recursos (assets) de la Asset Store creados por la comunidad de Unity.  En el proceso se han utilizado técnicas de metodología ágil, en concreto el método Kanban que nos permite un desarrollo fluido y entrega a tiempo limitando las cargas de trabajo en las diferentes etapas.  El resultado es un videojuego en su versión estable, perfectamente jugable y publicado en algunas de las principales plataformas del mercado.  Tablero de Trello: <https://trello.com/b/XbSeTA1e/adventure-of-deltaland-roadmap>  Proyecto en GitHub: <https://github.com/bernatrodriguez/The-Adventure-of-Deltaland>  Vídeo Avance 1: <https://youtu.be/jFlAeBX1TOw>  Vídeo Avance 2 <https://youtu.be/ETw1_MxZzNM> | |
| **Abstract:** | |
| The video game industry is a booming economic sector in which it is becoming easier and easier to get started. On the Internet we find numerous resources, among them: platforms, tutorials, assets, icons, textures, animations, audio effects, etc.  In this project we present the design and development process of a role-playing game (RPG) using Unity, one of the most used game engines nowadays by beginners due to its optimal learning curve and versatility. We also familiarize ourselves with the C# programming language and make use of other programs in the different stages of the design such as GIMP in the creation of textures or Audacity in the production of sound effects. We also use assets from the Asset Store created by the Unity community.  In the process we have used agile methodology techniques, specifically the Kanban method that allows a smooth development and delivery on time by limiting workloads at different stages.  The result is a video game in its stable version, perfectly playable and published in some of the main platforms of the market. | |

Agradecimientos

Este TFG es el resultado de estos últimos cuatro años de trabajo y aprendizaje. Quiero expresar mi agradecimiento a todas aquellas personas que creen y siempre han creído en mí. En primer lugar, a Roger, mi tutor, quien me ha guiado y aconsejado durante toda la carrera. En segundo lugar, quiero agradecer a mis padres, mi mujer y mi hija, su paciencia y apoyo incondicional sin los que nunca hubiese llegado hasta el final. Por último, quiero darle las gracias a mi compañero Héctor, por su apoyo y buenos consejos durante este tiempo. No podría haberlo hecho sin ti.

Abstract

La industria del videojuego es un sector económico que se encuentra en pleno apogeo y en el que cada vez es más sencillo iniciarse. En Internet encontramos numerosos recursos, entre ellos: plataformas, tutoriales, assets, iconos, texturas, animaciones, efectos de audio, etc.

En este proyecto se presenta el proceso de diseño y desarrollo de un videojuego de rol o RPG *(role-playing game*) mediante el uso de Unity, uno de los motores de videojuegos más utilizados hoy en día por principiantes debido a su óptima curva de aprendizaje y versatilidad. Asimismo, nos familiarizamos con el lenguaje de programación C# y hacemos uso de otros programas en las diferentes etapas del diseño tales como GIMP en la creación de texturas o Audacity en la producción de efectos de sonido. También utilizamos recursos (assets) de la Asset Store creados por la comunidad de Unity.

En el proceso se han utilizado técnicas de metodología ágil, en concreto el método Kanban que nos permite un desarrollo fluido y entrega a tiempo limitando las cargas de trabajo en las diferentes etapas.

El resultado es un videojuego en su versión estable, perfectamente jugable y publicado en algunas de las principales plataformas del mercado.

Palabras clave

*videojuego, RPG, unity*

Índice

[1. Introducción 11](#_Toc138005060)

[1.1. Un paseo por la historia de los videojuegos 11](#_Toc138005061)

[1.1.1. Genero role-playing game (RPG) 13](#_Toc138005062)

[1.1.2. Referencias 14](#_Toc138005063)

[1.1.3. Planteamiento inicial 14](#_Toc138005064)

[1.2. Descripción/Definición 16](#_Toc138005065)

[1.3. Objetivos generales 17](#_Toc138005066)

[1.3.1. Objetivos principales 17](#_Toc138005067)

[1.3.2. Objetivos secundarios 17](#_Toc138005068)

[1.4. Metodología y proceso de trabajo 18](#_Toc138005069)

[1.5. Planificación 20](#_Toc138005070)

[1.6. Presupuesto 21](#_Toc138005071)

[1.6.1. Recursos de software 21](#_Toc138005072)

[1.6.2. Recursos de hardware 21](#_Toc138005073)

[1.6.3. Recursos humanos 22](#_Toc138005074)

[1.7. Estructura del resto del documento 24](#_Toc138005075)

[2. Estado del arte 25](#_Toc138005076)

[2.1. Motores para el desarrollo de videojuegos en 2D 25](#_Toc138005077)

[2.1.1. Godot Engine 26](#_Toc138005078)

[2.1.2. Unity 26](#_Toc138005079)

[2.1.3. Unreal Engine 26](#_Toc138005080)

[2.1.4. Solar2D 27](#_Toc138005081)

[2.1.5. GamerMaker Studio 27](#_Toc138005082)

[2.1.6. RPG Maker 28](#_Toc138005083)

[2.1.7. Tabla comparativa 28](#_Toc138005084)

[2.1.8. Motor elegido para el desarrollo del proyecto 29](#_Toc138005085)

[2.2. Herramientas de desarrollo 30](#_Toc138005086)

[2.2.1. Visual Studio 30](#_Toc138005087)

[2.2.2. GIMP 30](#_Toc138005088)

[2.2.3. Audacity 31](#_Toc138005089)

[2.3. Plataformas de juego 31](#_Toc138005090)

[2.4. Distribución y monetización 32](#_Toc138005091)

[3. Propuesta 33](#_Toc138005092)

[3.1. Trama 33](#_Toc138005093)

[3.2. Objetivo 33](#_Toc138005094)

[3.3. Mapa 34](#_Toc138005095)

[3.4. Personajes 34](#_Toc138005096)

[3.4.1. Protagonista 34](#_Toc138005097)

[3.4.2. Amigos 34](#_Toc138005098)

[3.4.3. Enemigos 34](#_Toc138005099)

[4. Diseño 35](#_Toc138005100)

[4.1. Entorno de desarrollo escogido 35](#_Toc138005101)

[4.1.1. Interfaz 36](#_Toc138005102)

[4.1.2. Requerimientos de hardware 38](#_Toc138005103)

[4.2. Assets 38](#_Toc138005104)

[4.3. Arquitectura del proyecto 39](#_Toc138005105)

[4.4. Proceso de desarrollo 39](#_Toc138005106)

[4.4.1. Creación del proyecto 40](#_Toc138005107)

[4.4.2. Personaje principal 41](#_Toc138005108)

[4.4.3. Interfaz 41](#_Toc138005109)

[4.4.4. Mapa 44](#_Toc138005110)

[4.4.5. NPCs y diálogos 46](#_Toc138005111)

[4.4.6. Lootdrop 47](#_Toc138005112)

[4.4.7. Misiones 48](#_Toc138005113)

[4.4.8. Combate y puntos de experiencia (XP) 48](#_Toc138005114)

[4.4.9. Comercio y crafting 49](#_Toc138005115)

[5. Implementación 51](#_Toc138005116)

[5.1. Requisitos de instalación 51](#_Toc138005117)

[5.2. Instrucciones de instalación 51](#_Toc138005118)

[6. Demostración 52](#_Toc138005119)

[6.1. Controles del juego 52](#_Toc138005120)

[6.2. Tests 52](#_Toc138005121)

[6.3. Guía de usuario 53](#_Toc138005122)

[7. Conclusiones y líneas de futuro 55](#_Toc138005123)

[7.1. Conclusiones 55](#_Toc138005124)

[7.2. Líneas de futuro 55](#_Toc138005125)

[Glosario 57](#_Toc138005126)

[Bibliografía 58](#_Toc138005127)

Figuras y tablas

Índice de figuras

[Figura 1: Spacewar! en el Museo Histórico de Ordenadores 11](#_Toc138005025)

[Figura 2: DND, el primer RPG de la historia 12](#_Toc138005026)

[Figura 3: Nivel 1 del videojuego DOOM 12](#_Toc138005027)

[Figura 4: To the Moon 14](#_Toc138005028)

[Figura 5: Estilo visual 15](#_Toc138005029)

[Figura 6: Motores de videojuegos 25](#_Toc138005030)

[Figura 7: Videojuegos creados con Unity 30](#_Toc138005031)

[Figura 8: Steam, la mayor plataforma de distribución digital de videojuegos 32](#_Toc138005032)

[Figura 9. Logotipo de Unity 35](#_Toc138005033)

[Figura 10. Interfaz de Unity 36](#_Toc138005034)

[Figura 11. Logotipo de la Unity Asset Store 39](#_Toc138005035)

[Figura 12. Nuevo Proyecto de Unity 40](#_Toc138005036)

[Figura 13. Renderizador 40](#_Toc138005037)

[Figura 14. Animaciones 41](#_Toc138005038)

[Figura 15. Blend Tree 41](#_Toc138005039)

[Figura 16. Interfaz del juego 42](#_Toc138005040)

[Figura 17. Panel de Stats 42](#_Toc138005041)

[Figura 18. Panel de Inventario 43](#_Toc138005042)

[Figura 19. Panel de Quests activas 43](#_Toc138005043)

[Figura 20. Mapa de Deltaland 45](#_Toc138005044)

[Figura 21. Tiling Rules 46](#_Toc138005045)

[Figura 22. Decoraciones 46](#_Toc138005046)

[Figura 23. Diálogos 47](#_Toc138005047)

[Figura 24. Ejemplo de lootdrop de un enemigo 47](#_Toc138005048)

[Figura 25. Panel de Quests del Inspector 48](#_Toc138005049)

[Figura 26. Combate 49](#_Toc138005050)

[Figura 27. Panel de la tienda 50](#_Toc138005051)

[Figura 28. Panel de crafting 50](#_Toc138005052)

[Figura 29. Controles del juego 52](#_Toc138005053)

[Figura 30. Tests de seguridad 53](#_Toc138005054)

Índice de tablas

[Tabla 1. Fechas clave 21](#_Toc138005055)

[Tabla 2. Diagrama de Gantt 21](#_Toc138005056)

[Tabla 3. Presupuesto 24](#_Toc138005057)

[Tabla 4. Arquitectura del proyecto 40](#_Toc138005058)

[Tabla 5. Valores de rendimiento 54](#_Toc138005059)

# Introducción

## Un paseo por la historia de los videojuegos

Ya hace tiempo que los videojuegos dejaron de ser una siempre afición para convertirse en una de las industrias que más factura en el mundo, pero es importante que sepamos cómo ocurrió todo, cuál fue su origen y su gran escalada que permitió que los videojuegos fueran lo que a día de hoy son.

Debido a las múltiples definiciones que podemos darle al concepto videojuego, es realmente difícil señalar cuándo fue el primer videojuego de la historia. Todo comienza con una serie de sucesos en diferentes lugares y momentos de la historia, no obstante, podríamos destacar dos grandes acontecimientos y es cuando en 1952, Alexander S. Douglas creó “OXO”, una versión computarizada del tres en raya que enfrentaba al jugador contra la máquina; y en 1958, William Higginbotham logra jugar “Tennis for two” en un osciloscopio. Lo cierto es que como este juego era técnicamente una representación de señales eléctricas, no fue hasta 1962 cuando Steve Russell creó “Spacewar!”, el considerado **primer videojuego de la historia**. Este consistía en dos naves espaciales, «la aguja» *(the needle*) y «la cuña» (*the wedge*), controladas por jugadores diferentes, y que intentan dispararse entre sí en torno al campo gravitacional de una estrella.

Imagen de la pantalla de un computador

Descripción generada automáticamente con confianza baja

Figura 1: Spacewar! en el Museo Histórico de Ordenadores

En 1972 es fundada la empresa Atari, y Nolan Bushnell junto con Allan Alcorn crean “Pong”, el **primer videojuego comercial** de la historia, el cual estaba basado en el deporte de tenis de mesa. Posteriormente sucedieron otros videojuegos como **”The Game of Dungeons” (“DND”)** en 1974, **el primer RPG** de la historia, basado en el juego de mesa “Dungeons & Dragons”, y otros conocidos clásicos como “Space Invaders” en 1978, “Asteroids” en 1979 o Pac-Man en 1980.

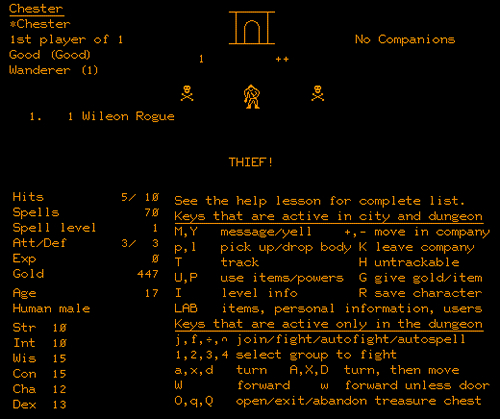


Figura 2: DND, el primer RPG de la historia

En 1981, tratando de hacer competencia con Pac-Man, Shigeru Miyamoto crea Donkey-Kong, en el que un gran simio secuestra a una princesa, la novia de Jumpman (más tarde conocido como Mario Bros).

En 1985 se lanza al mercado nipón la Nintendo Entertainment System (NES) con el gran éxito de Super Mario Bros. Años después comienza la rivalidad entre Mario Bros y Sonic con el lanzamiento de la Super Nintendo Entertainment System (SNES) y la Sega Genesis.

En 1993 sale al mercado DOOM, el considerado primer videojuego de disparos en primera persona y uno de los primeros en 3D, que marcó un antes y un después en la historia de los videojuegos.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Figura 3: Nivel 1 del videojuego DOOM

Los videojuegos formados por polígonos comienzan a ganar éxito con la primera PlayStation en 1994 y se inicia la guerra de las consolas junto con la Sega Saturn. En 1999 esta misma compañía lanza la Sega Dreamcast que permite por primera vez el juego en línea en consolas y en el 2000 Sony lanza la PlayStation 2, la consola más vendida de la historia. Microsoft se incorpora posteriormente a la industria con el lanzamiento de la Xbox en 2001.

En 2006 llegan la PlayStation 3, la Xbox 360 y la Nintendo Wii con un gran salto técnico en cuanto a realismo gráfico y jugabilidad. Los videojuegos se popularizan de forma masiva y ya son una de las mayores industrias del momento. En 2009 estos llegan a los teléfonos móviles y son accesibles para la mayoría de las personas, los videojuegos ya son una gran parte de nuestras vidas.

Posteriormente llegarán otros hitos como la octava generación (PlayStation 4, Xbox One y Nintendo Wii U), la realidad virtual, el juego en streaming y los deportes electrónicos o e-Sports. Estaba demostrado, los videojuegos habían venido para quedarse.

### Genero role-playing game (RPG)

Los RPG son un tipo de videojuego que se caracteriza por incluir una gran variedad de estilos de juego mediante mecánicas como el desarrollo de estadísticas del personaje. Algunos de ellos provienen de juegos de mesa de rol como Dungeons & Dragons.

Generalmente la mayoría de los videojuegos de este género comparten algunas características comunes en su jugabilidad:

* **Historia:** tienen una narrativa compleja, con personajes detallados y una trama profunda. El jugador tiene la libertad de explorar el mundo y tomar decisiones que afectan al trascurso del propio videojuego.
* **Combate:** incluyen un sistema de combate, que puede ser por turnos o en tiempo real. El jugador puede usar habilidades especiales del personaje para derrotar a los enemigos.
* **Progresión:** tienen un sistema de progresión que permite al jugador mejorar las habilidades de su personaje según avanza en el juego. Esto puede incluir subir niveles de experiencia, obtener nuevos equipos y adquirir habilidades especiales.
* **Mundo abierto:** presentan un mundo abierto para explorar, con múltiples ubicaciones y áreas para descubrir. El jugador tiene la libertad de explorar el mundo a su propio ritmo y descubrir secretos o misiones secundarias.

Existen numerosos tipos de RPG, cada uno con sus propias características, algunos de los más influyentes son los RPG japoneses (JRPG), los occidentales (WRPG), los de multijugador masivos (MMORPG), los de acción (Action RPG) y los roguelike, que se caracterizan por su alta dificultad y sus niveles generados de forma aleatoria.

Algunos de los RPG más famosos de la historia son World of Warcraft (2004), Pillars of Eternity (2011), Neverwinter Nights (2002) o las sagas Final Fantasy, The Witcher. The Elder Scrolls y Dark Souls.

### Referencias

**To the Moon:** se centra en la historia de dos médicos que intentan cumplir el último deseo de un hombre a punto de morir: ir a la luna.

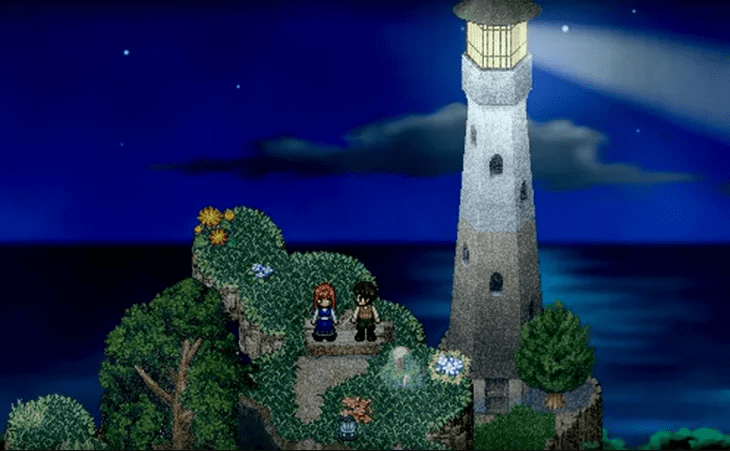


Figura 4: To the Moon

**Corpse Party:** un juego de terror centrado en un grupo de estudiantes que quedan atrapados en una escuela después de realizar un ritual mágico.

**Yume Nikki**: una chica llamada Madotsuki explora un mundo de sueños extraños y perturbadores.

**OneShot:** juego de aventuras en el que un niño llamado Niko debe recuperar su sol, que murió hace mucho tiempo.

### Planteamiento inicial

Mi idea inicial, ya que es mi primer videojuego, es crear un RPG clásico “tipo RPG Maker” pero utilizando un motor más avanzado como es Unity. Me gustaría implementar una pequeña historia y añadir las mecánicas más típicas de este tipo de videojuegos como es tener un inventario en el que guardar objetos, tener conversaciones con NPCs, enemigos, combate, un sistema básico de misiones y un método para comerciar con objetos.

El estilo visual sería lo más parecido posible a la siguiente imagen:



Figura 5: Estilo visual

## Descripción/Definición

Es obvio que los videojuegos cuentan con un papel muy relevante en nuestra sociedad, es una gran herramienta económica y que cubre las necesidades de ocio de muchas personas de nuestro planeta. Actualmente la industria del videojuego supera incluso a la cinematográfica debido a su gran capacidad para generar ingresos a corto, medio y largo plazo. Puede ser una inversión de tiempo y dinero con vistas al futuro, a ser soportado, adaptado y actualizado durante una gran cantidad de años. Algunos ejemplos pueden ser los archiconocidos Dota 2 (2013) o World of Warcraft (2004), todavía jugados a día de hoy por gran cantidad de personas.

Es por esto que este proyecto apuesta por mostrar el desarrollo de lo que sería un videojuego, pero concretamente del género *role-playing game* (RPG), uno de los más versátiles y jugados a día de hoy. Es relativamente sencillo desarrollarse por una sola persona gracias a los últimos motores gráficos que facilitan enormemente las tareas, algo que hace veinte años sería imposible de imaginar.

Una de las grandes ventajas de desarrollar en este género es que se pueden crear niveles e implementar mecánicas en una fase previa del desarrollo sin necesidad de incorporar historia o elementos decorativos de nivel. Esta flexibilidad es una de las grandes motivaciones que me han llevado a embarcarme en este proyecto.

## Objetivos generales

Listado y descripción de los objetivos del TF, ordenados por relevancia.

### Objetivos principales

Objetivos de la aplicación/producto/servicio:

* Desarrollo de un videojuego completo, que incluya:
  + Texturas y modelos 2D: personajes, objetos, escenarios, etc.
  + Efectos de sonido y música.
  + Interfaz de Usuario Gráfica (GUI).
  + Programación de diferentes scripts para incluir diferentes mecánicas.
  + Historia que ponga en contexto al videojuego.
* Posicionarse y darse a conocer en Internet como una base para una versión mejorada y comercializada del proyecto.
* Servir de inspiración para otros proyectos.

Objetivos para el cliente/usuario:

* Jugable en diferentes plataformas de escritorio, de forma que cada usuario tenga la posibilidad de jugar en su favorita y de la forma más cómoda posible (esto incluye controles alternativos).
* Transmitir un mensaje al jugador mediante la narración de la historia, al igual que lo hace una película.

Objetivos personales del autor del TF:

* Aprender a crear un videojuego desde cero (sin el uso de una plantilla) con un motor de videojuegos (Unity), comprendiendo así los requerimientos y el proceso de diseño y desarrollo.
* Enriquecer mis conocimientos informáticos en base al uso de programas complementarios al desarrollo del videojuego, como sería Visual Studio, GIMP y Audacity.
* Prepararme para la creación de futuros nuevos proyectos, de forma que pueda dedicarme profesionalmente a ello.

### Objetivos secundarios

Objetivos adicionales que enriquecen el TF.

* Añadir pequeños secretos o “*easter eggs*”, que añadirán guiños o valor adicional a la historia cuando el jugador explore intensamente el videojuego.

## Metodología y proceso de trabajo

Gracias a la inmensidad del sector y la gran cantidad de herramientas con las que contamos a día de hoy, en el desarrollo de videojuegos contamos con una gran variedad de metodologías ágiles que nos permiten poder llevar nuestro proyecto al éxito. Algunas de ellas son:

* **Extreme Programming (XP)**

Esta metodología trata de crear una relación estrecha entre el equipo de programación y el cliente, creando versiones del juego con una elevada frecuencia, generalmente cada dos semanas. Esto quiere decir que el diseño final no está definido, sino que se adapta sobre las necesidades de cada momento. Las versiones se van modificando constantemente hasta obtener el resultado deseado.

* **Scrum**

Se utiliza principalmente en entornos colaborativos y complejos, en los que se desea obtener un resultado temprano y los requisitos no están perfectamente definidos. Esta metodología fomenta la competitividad y la innovación.

* **Open Source**

Estos proyectos se caracterizan por distribuir las cargas de trabajo entre diferentes equipos, contando con un solo supervisor que será el encargado de realizar los cambios en el código fuente. Es una metodología en la que la depuración del código es distribuida entre muchas personas por lo que es recomendable para proyectos problemáticos.

* **Kanban**

Esta metodología es ideal para distribuir las cargas de trabajo entre los miembros del equipo según su disponibilidad y habilidades, además se basa en la mejora continua del videojuego.

Esta se ejecuta mediante tableros Kanban, un método que permite mostrar de manera visual la carga de trabajo. Este tablero se divide en columnas, cada una representando una etapa del proyecto y por las cuales moveremos las diferentes tareas. Generalmente encontramos el tablero dividido en tres columnas: *To do* (pendiente), *In progress* (en progreso) y *Done* (terminado).

Gracias a que es una metodología ágil, adaptativa a cada circunstancia, con un desarrollo evolutivo y de entrega temprana, **en este proyecto vamos a utilizar la metodología Kanban** mediante un tablero de Trello.

Enlace al tablero: [Adventure of Deltaland](https://trello.com/b/XbSeTA1e/adventure-of-deltaland-roadmap)

El proceso de trabajo se realizará utilizando el motor de videojuegos Unity, uno de los más sencillos para principiantes y que a su vez ofrece un gran potencial.

Con el uso de GIMP crearemos y modificaremos algunas texturas para añadirlas posteriormente al juego y con Audacity crearemos algunos efectos de sonido. Por supuesto también haremos uso de recursos de terceros como una banda sonora de una biblioteca de sonidos.

Programaremos scripts en C# con Visual Studio para añadir funcionalidades al juego y crearemos una interfaz sencilla e intuitiva.

Todo este proceso será llevado a cabo mediante la metodología ágil de Kanban.

## Planificación

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **FECHA CLAVE** | | |
|  |  | **Comienzo** | **Fin** | **Descripción** |
| **PEC** | **1** | 01/03/2023 | 12/03/2023 | Plan de Proyecto |
| **2** | 13/03/2023 | 17/04/2023 | Estado del arte y primera versión del proyecto |
| **3** | 18/04/2023 | 21/05/2023 | Implementación de versión jugable |
| **4** | 22/05/2023 | 18/06/2023 | Memoria y productos finales |
|  | 19/06/2023 | 27/06/2023 | Semana de receso |
| **5** | 28/06/2023 | 05/07/2023 | Defensa Virtual – Tribunal de TF |

Tabla 1. Fechas clave

**Diagrama de Gantt**

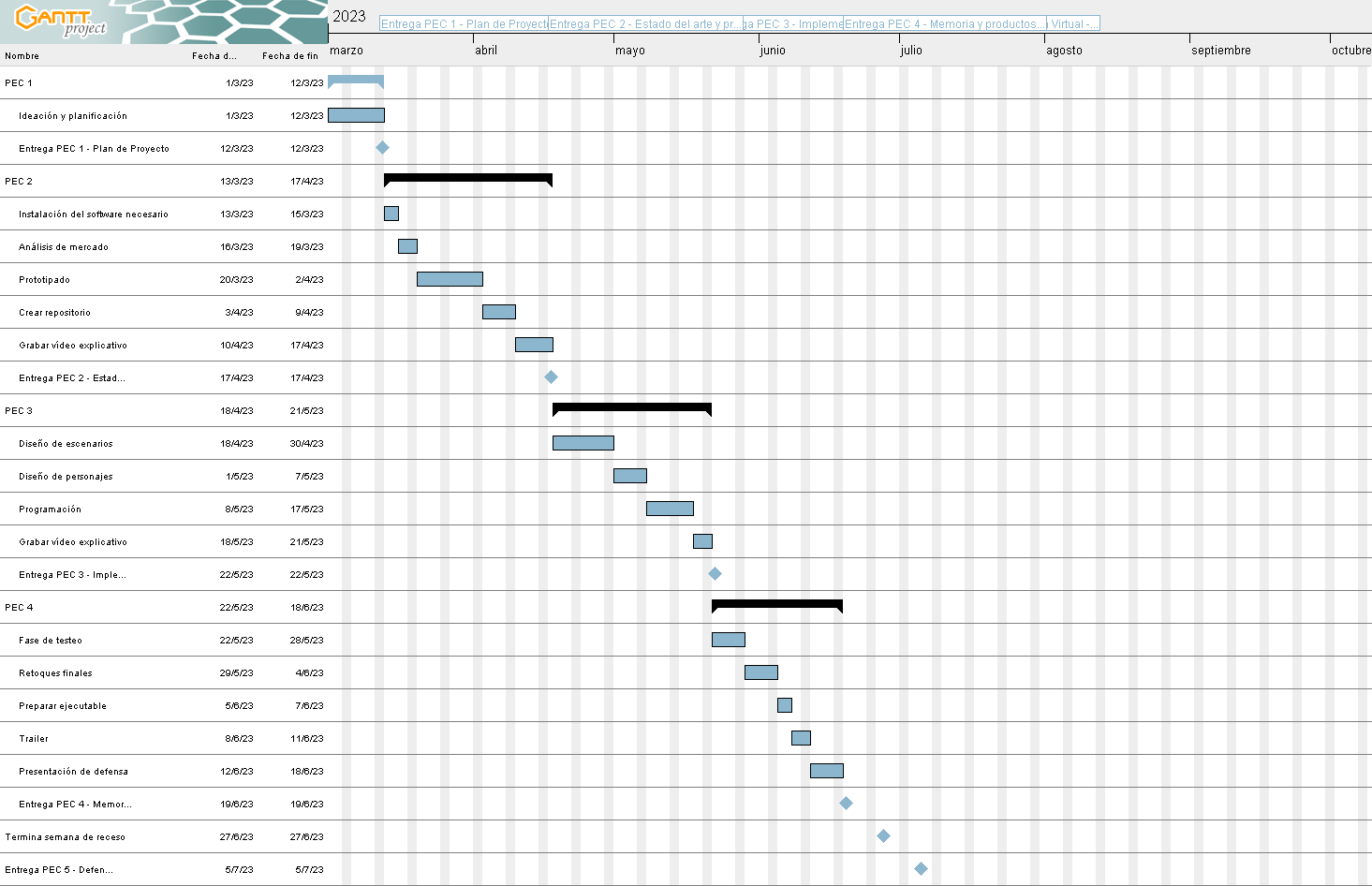
****

Tabla 2. Diagrama de Gantt

## Presupuesto

### Recursos de software

|  |  |  |
| --- | --- | --- |
| **RECURSOS DE SOFTWARE** | | |
| **Acción** | **Descripción** | **Coste** |
| Unity | Motor de videojuegos  Licencia Personal (Gratuita) | 0 $ |
| Visual Studio | IDE para la programación del videojuego | 0 $ |
| GIMP | Editor de imágenes avanzado | 0 $ |
| Audacity | Editor de audio avanzado | 0 $ |
| Plugins | Funciones adicionales para el motor de videojuegos | 50 $ |
| Modelos y texturas | Comprados en la Asset Store (p. ej. Vegetación) | 50 $ |
| Efectos visuales | Comprados en la Asset Store (p. ej. Explosión) | 50 $ |
| Efectos de sonido / Música | Comprados en la Asset Store (p. ej. Animales) | 50 $ |
| Tipografías | Gratuitas y de uso libre | 0 $ |
| **TOTAL SOFTWARE** | | **200 $** |

### Recursos de hardware

|  |  |  |
| --- | --- | --- |
| **RECURSOS DE HARDWARE** | | |
| **Acción** | **Descripción** | **Coste** |
| Ordenador |  | 1.999 $ |
| Monitores externos |  | 569 $ |
|  | 249 $ |
| Disco duro de 1TB | Almacenamiento para los archivos de desarrollo del videojuego. | 49 $ |
| **TOTAL HARDWARE** | | **2866 $** |

### Recursos humanos

|  |  |  |
| --- | --- | --- |
| **RECURSOS HUMANOS** | | |
| **Acción** | **Descripción** | **Coste** |
| Programador | Desarrollador especializado en el desarrollo de videojuegos.  *Nota: en este caso el coste real es de 0 $ ya que yo mismo me hago cargo del desarrollo del videojuego.* | 15 $ / hora  300 horas  TOTAL: 4500 $ |
| **TOTAL RRHH** | | **4500 $** |

|  |  |  |
| --- | --- | --- |
| **OTROS RECURSOS** | | |
| **Acción** | **Descripción** | **Coste** |
| Marketing | Publicidad para el producto | 100 $ |
| Licencias | Tarifa de Steam Direct | 100 $ |
| **TOTAL OTROS** | | **200 $** |

|  |  |
| --- | --- |
| **TOTAL** | **7766 $** |

Tabla 3. Presupuesto

## Estructura del resto del documento

* Estado del arte: se analizan detalladamente los diferentes motores de videojuegos que podemos encontrar en el mercado, así como el género RPG, videojuegos que han servido de inspiración y plataformas para las cuales deberíamos desarrollar.
* Propuesta: se especifican los objetivos del proyecto, así como el modelo de negocio y la estrategia de marketing que seguirá.
* Diseño: se explica todo el proceso de elaboración de las diferentes partes del proyecto, así como la estructura que sigue el mismo.
* Implementación: consiste en los requisitos recomendados e instrucciones de instalación del videojuego.
* Demostración: este apartado incluye los controles del juego, los diferentes test que se realicen y la guía de usuario.
* Conclusiones y líneas de futuro: se muestran las conclusiones personales en base a los resultados del proyecto, así como detalles sobre las próximas mejoras a realizar de cara al futuro.
* Glosario: en este apartado se incluyen las palabras fuera del lenguaje común que se han ido utilizando a lo largo del documento.
* Bibliografía: contenido multimedia, referencias y publicaciones mencionadas en el documento.

# Estado del arte

## Motores para el desarrollo de videojuegos en 2D

Los motores de videojuegos (en inglés *game engines*) son un tipo de software especialmente dedicado al diseño y desarrollo de un videojuego. Proporcionan el conjunto de herramientas necesarias en una sola interfaz para facilitar al máximo el desarrollo del producto.



Figura 6: Motores de videojuegos

Cuando queremos elegir un motor de videojuegos debemos analizar algunas de sus características:

* **Precio y comisiones:** los hay completamente gratuitos, algunos cobran una comisión a partir de ciertos beneficios y otros funcionan mediante una suscripción mensual.
* **Curva de aprendizaje:** quizás sea el parámetro más importante, si estamos comenzando en el mundo del desarrollo de videojuegos deberemos elegir un software con una curva poco pronunciada.
* **Motor gráfico:** es el encargado de calcular y mostrar las texturas, los polígonos y la iluminación en pantalla, un motor gráfico avanzado permitirá un mayor realismo en el videojuego.
* **Motor físico:** controla el comportamiento físico del videojuego, es decir, realiza cálculos sobre parámetros como la masa, el volumen, la velocidad o la aceleración de un objeto.
* **Motor de sonido:** carga las pistas de sonido y con los parámetros necesarios, pudiendo aplicar distorsiones o comunicarse con el motor físico para producir, por ejemplo, el efecto Doppler.
* **Plataformas de juego:** cada motor cuenta con soporte a ciertas plataformas a las que podemos exportar el videojuego, no todos tienen las mismas.

### Godot Engine

Este motor de videojuegos es *open source* y completamente gratuito, por lo que en este aspecto es insuperable. Cuenta con una amplia documentación para que nos resulte sencillo el aprendizaje de este y la curva de aprendizaje sea lo más baja posible.

Tiene un motor gráfico bastante capaz (aunque lejos de un aspecto profesional) y que avanza continuamente gracias a la comunidad de desarrollo del motor. Tanto el motor físico como el de sonido son excelentes y da soporte a las plataformas de juego más utilizadas: Windows, macOS, Linux, iOS, Android, web y consolas.

Una ventaja que nos ofrece respecto a otros motores es que en *scripting* permite el uso de varios lenguajes de programación, pero también tiene un gran inconveniente y es que no nos ofrece una tienda de recursos adicionales.

### Unity

Este motor va un paso más allá, ya que nos permite de forma relativamente sencilla obtener resultados profesionales. Una gran ventaja de Unity es que cuenta con una gran comunidad de usuarios por lo que encontramos fácilmente documentación y foros donde aprender y buscar ayuda para nuestro desarrollo.

En cuanto a precios, Unity se basa en los ingresos que tengamos utilizando el software. Siempre que no superemos los 100.000 $ al año podremos utilizar Unity Personal, que es completamente gratuito. A partir de esta cifra encontramos opciones de suscripción (Plus, Pro y Enterprise).

Es el motor más utilizado para el desarrollo de videojuegos para plataformas móviles, aunque también es muy utilizado en PC y soporta más de 25 plataformas. Su lenguaje de programación es C#.

Unity nos ofrece la **Asset Store**, donde podemos encontrar todo tipo de recursos como modelos 2D, texturas, efectos, scripts o sonidos que han creado otras personas. Los hay gratuitos y de pago.

### Unreal Engine

Las capacidades de Unreal mejoran el desarrollo de videojuegos 3D, pero no es el caso para los videojuegos en 2D, ya que de forma nativa no nos ofrece herramientas para ello por lo que debemos hacer uso siempre de herramientas externas o *plugins*. La documentación que nos ofrece Unreal es igual de completa que en Unity, pero no contamos con una comunidad tan grande con la que colaborar y aprender. Su curva de aprendizaje es algo más alta que en Unity, sobre todo para el desarrollo en dos dimensiones.

Este motor cobra una comisión del 5% cuando los ingresos superan los 3.000 $ al trimestre, por debajo de esta cifra es completamente gratuito.

Actualmente soporta 18 plataformas de juego, y utiliza los lenguajes de programación C++ y Blueprints, un lenguaje propio muy visual y sencillo para las personas que se inician en el desarrollo de videojuegos.

La **Unreal Marketplace** nos ofrece todo tipo de recursos al igual que la Asset Store, y encontramos tanto recursos gratuitos como de pago.

### Solar2D

Este motor basado en Lua es de los más sencillos que podemos encontrar a día de hoy para el desarrollo de videojuegos 2D, cabe remarcar que es un proyecto *open source* por lo que es completamente gratuito. Ya que proviene del archiconocido Corona SDK, podemos encontrar suficiente documentación en la web, aunque su comunidad de usuarios a día de hoy no es demasiado grande.

Es un motor multiplataforma que soporta los sistemas operativos más utilizados: iOS, tvOS, Android, Android TV, Windows, Linux y HTML5.

Aunque no cuenta con una tienda de assets oficial, sí que cuenta con numerosos plugins y compatibilidad con tiendas de terceros.

### GameMaker Studio

En sus inicios fue conocido como Animo y es un poderosísimo motor gráfico 2D que no necesita programación gracias a su sencillo sistema de eventos, también cuenta con un gran motor de físicas y varios comportamientos prestablecidos. No cuenta con una gran comunidad ni documentación, pero su curva de aprendizaje es muy rápida por lo que es ideal para un principiante.

Es un motor de uso gratuito en su modalidad básica, pero también ofrece tres versiones de pago que nos permiten hacer exportaciones a diferentes plataformas, por ejemplo, si no pagamos no podremos exportar nuestro videojuego a Windows o Android, que son las plataformas más utilizadas.

Permite el desarrollo para escritorio, web, móvil y consolas, sumando 17 plataformas en total. Es la opción menos versátil, aunque ofrece grandes posibilidades. Para *scripting* (solo en proyectos más avanzados) admite su propio lenguaje de programación *Game Maker Language* (GML).

No cuenta con una plataforma de recursos creados por la comunidad como si lo hacen Unity o Unreal Engine.

### RPG Maker

Al igual que el anterior, RPG Maker es un sencillo motor, en este caso enfocado al género RPG, que nos permite crear videojuegos sin tener nociones de programación. Se caracteriza por incluir numerosos materiales como mapas, personajes, iconos, música o efectos de sonido. Gracias a sus posibilidades, tiene una gran comunidad lo que se traduce en recursos y documentación que nos pueden ayudar a la hora de realizar nuestro videojuego.

Este motor tiene muchas versiones disponibles en el mercado y son de pago, pero podemos encontrarlas por precios realmente asequibles y no deberemos de pagar ningún tipo de suscripción.

La última versión disponible, RPG Maker MV permite exportar los juegos a PC, Mac, iOS, Android, web y Nintendo Switch. Nos permite *scripting* mediante JavaScript, pero es algo totalmente opcional.

Al igual que GameMaker Studio, no cuenta con una tienda de recursos creados por la comunidad ya que se basa en utilizar los propios recursos que encontramos en el motor.

### Tabla comparativa

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Motor 2D | Godot Engine | Unity | Unreal Engine | Solar2D | GameMaker Studio | RPG Maker |
| Precio | Gratis | Gratis (Personal)  399 $ / año (Plus)  2.040 $ / año (Pro)  Enterprise | Gratis  5% de los ingresos (a partir de 3.000 $ / trimestre) | Gratis | Gratis (Free)  49 $ / año (Creator)  99 $ / año (Indie)  799 $ / año (Enterprise) | De pago (el precio puede variar dependiendo de la versión) |
| Documentación | Buena | Excelente | Excelente | Escasa | Escasa | Buena |
| Comunidad | Mediana | Muy grande | Grande | Pequeña | Mediana | Grande |
| Curva de aprendizaje | Baja | Baja - Media | Media | Baja | Baja | Baja |
| Plataformas soportadas | +10 | +25 | 18 | 7 | 17 | 6 |
| Scripting | GDScript, Visual Scripting, C# y C++ | C# | C++ y Blueprints | Lua | GML | JavaScript |
| Tienda de recursos |  | Asset Store | Unreal Marketplace | Tiendas de terceros |  |  |

### Motor elegido para el desarrollo del proyecto

Viendo la anterior tabla comparativa, para un primer proyecto podemos descartar Unreal Engine, y quedarnos con Godot Engine, Unity, Solar2D, GameMaker Studio y RPG Maker. Vamos a analizarlos uno por uno:

* Godot Engine: es gratuito y sencillo de utilizar, pero no dispone de una gran comunidad que podría sernos de mucha utilidad para un primer videojuego.
* Unity: cuenta con una licencia gratuita, una muy buena documentación y una gran comunidad. También es el equilibrio perfecto entre facilidad de uso, soporte y potencial. Dispone de una de las tiendas de recursos más grandes del sector.
* Solar2D: aunque también es gratuito y fácil de aprender, su escasa documentación y pequeña comunidad a día de hoy hace que no sea una opción viable.
* GameMaker Studio: cuenta con una licencia gratuita, pero esta no nos permite exportar el videojuego en las plataformas principales por lo que no videojuegos distribuir el proyecto si no pagamos.
* RPG Maker: a primera vista podríamos decir que es el motor ideal para el desarrollo de este proyecto, ya que se enfoca principalmente en los videojuegos RPG, pero personalmente prefiero un motor que nos permita de una forma más libre el *scripting* y podamos desarrollar como lo haríamos para cualquier otro tipo de videojuego.

Es, por tanto, que el motor elegido para el desarrollo de este proyecto es **Unity**.



Figura 7: Videojuegos creados con Unity

## Herramientas de desarrollo

La creación de videojuegos es un proceso complejo que trata diferentes etapas, desde la conceptualización y diseño hasta la programación y desarrollo de los elementos del juego. Para esto es necesario utilizar una serie de herramientas y programas.

A continuación, detallaremos las utilizadas para la creación de The Adventure of Deltaland:

### Visual Studio

Se trata de un entorno de desarrollo integrado (IDE) utilizado para desarrollar aplicaciones de software y se integra fácilmente con el motor de videojuegos Unity. Nos ofrece numerosas características, como un depurador, un editor de código fuente, un diseñador de interfaces gráficas de usuario, y compatibilidad con diferentes lenguajes de programación como C++, C#, Java y Python. En el desarrollo de videojuegos, se puede utilizar para programar el juego, compilarlo y depurarlo.

### GIMP

Es un software de edición de imágenes *open source* y gratuito que cuenta con características avanzadas de edición como herramientas de selección, infinidad de pinceles, capas y filtros, y es compatible con varios formatos de imagen como PNG, JPEG, GIF y BMP. En el desarrollo de videojuegos es adecuado para la creación de *sprites*, texturas y otros elementos visuales que se utilizan en el juego.

### Audacity

Es un software de grabación y edición de audio *open source* y gratuito que nos ofrece herramientas de grabación de audio y herramientas de edición de audio multipista, como cortar, copiar y pegar, y efectos de sonido como amplificar, distorsión, eco y eliminación de ruido. En el desarrollo de videojuegos podemos usarlo para grabar y editar tanto efectos de sonido como música para el juego.

En resumen, Visual Studio es la herramienta de programación que nos permitirá realizar el scripting, GIMP es una herramienta de edición de gráficos que nos permitirá crear y modificar texturas y Audacity es una herramienta de edición de audio con la que crearemos efectos de sonido y música para el videojuego. Los tres programas son importantes en la creación de videojuegos y se utilizan para diferentes aspectos del proceso de desarrollo.

## Plataformas de juego

Según el tipo de videojuego, tiene una forma de interacción diferente y por ello lo hace más jugable en unas plataformas que en otras.

Ya que “The Adventure of Deltaland” se trata de un RPG, es decir, un videojuego de rol, necesitaremos controles precisos como teclado + ratón o mando, no tendría mucho sentido controlarlo con por ejemplo una pantalla táctil.

Cabe mencionar que se han visto muy buenas adaptaciones de RPG en dispositivos móviles con controles adaptados, como pueden ser el famoso “Genshin Impact” o el mítico “Albion Online”, pero la jugabilidad de los controles virtuales nunca es comparable con la que nos ofrece un control físico como el de una consola o un ordenador, que en acciones como recoger los objetos de un cofre puede suponer una gran diferencia.

Las consolas también tienen un gran inconveniente, y es que cada videojuego requiere de ciertas optimizaciones para maximizar el rendimiento en un hardware concreto, esto requiere invertir mucho tiempo en la fase de testeo y es algo que no podemos permitirnos en este proyecto.

Descartando los dispositivos móviles por los controles virtuales y las consolas por sus requerimientos de optimización, vamos a quedarnos con las tres principales plataformas de escritorio: Windows, macOS y Linux.

## Distribución y monetización

Para distribuirlo de forma gratuita en las tres mencionadas plataformas de escritorio, utilizaremos la mayor y más popular tienda de distribución digital de videojuegos, Steam.



Figura 8: Steam, la mayor plataforma de distribución digital de videojuegos

Como desarrolladores, Steam nos proporciona una amplia audiencia, herramientas de promoción, facilidad de distribución y buena gestión de actualizaciones y soporte.

Por supuesto, también ofrece grandes ventajas al jugador:

* Instalación y actualización automática del juego
* Comunidad del juego y transmisiones de vídeo
* Descuentos y ofertas especiales
* Steam Cloud (guardado en la nube)
* Steam Link (posibilidad de jugar en remoto)
* Soporte técnico y foros de discusión

# Propuesta

## Trama

La trama del videojuego The Adventure of Deltaland sigue las aventuras de Ansel, un niño que se embarca en una emocionante búsqueda del tesoro.

La historia comienza con Ansel descubriendo un antiguo mapa del tesoro en el ático de la casa de sus fallecidos abuelos. Decidido a descubrir el misterio detrás del mapa, el protagonista abandona su hogar y decide emprender un viaje a la isla de Deltaland en busca del tesoro escondido.

Durante su búsqueda, Ansel descubre que no es el único que está en busca del tesoro, y debe luchar contra otros buscadores de tesoros y extrañas criaturas, resolver acertijos y superar peligrosas trampas para poder avanzar.

A medida que se acerca al tesoro, también descubre la historia detrás de él, incluyendo los peligros que enfrentó su abuelo para ocultarlo. Ansel debe confiar en sus habilidades y luchar para lograr su objetivo.

El videojuego es una mezcla de aventuras, puzles y combate, con Ansel como protagonista y con habilidades únicas que debe utilizar para resolver los diferentes desafíos.

En resumen, The Adventure of Deltaland es un videojuego emocionante y divertido para todas las edades, que destaca los valores del amor por la familia, la lucha por lo que queremos y la aventura. Los jugadores podrán disfrutar de una emocionante trama, encontrando pistas y descubriendo los misterios detrás del tesoro.

La trama será contada al jugador mediante textos en objetos y los diálogos con los personajes del videojuego.

## Objetivo

El objetivo del protagonista es encontrar el tesoro que dejó su abuelo en la isla de Deltaland. Para lograr esto, Ansel deberá explorar la isla, superar desafíos y resolver puzles para descubrir la ubicación del tesoro. Durante su aventura, tiene que enfrentarse a las criaturas que habitan la isla, lo que añade un elemento de peligro y emoción al juego. El éxito del protagonista en su búsqueda del tesoro dependerá de su habilidad para evitar obstáculos, combatir enemigos y encontrar pistas que lo lleven al tesoro final.

El videojuego seguirá una historia principal y una serie de misiones secundarias o *quests*, pero en un mundo abierto, por lo que contra más explore el personaje más nos envolveremos en ella.

## Mapa

El paradisiaco lugar donde se ubica el videojuego es una pequeña isla del Pacífico llamada Deltaland, está habitada por humanos que residen en pequeñas aldeas, pero también existen extrañas criaturas autóctonas de la isla.

Repartidos por el mapa, podemos encontrar diferentes diálogos y secretos escondidos que revelarán pistas sobre la ubicación del tesoro, así como objetos que nos sirvan en el combate contra enemigos.

## Personajes

### Protagonista

El personaje principal de este videojuego es el joven Ansel, es valiente y decidido, y está dispuesto a enfrentar cualquier desafío para cumplir con su objetivo. Siempre ha tenido una personalidad curiosa y aventurera, y disfruta al explorar y descubrir cosas nuevas.

Ansel desea honrar el legado de su abuelo ya que siempre sintió una conexión especial con su fallecido abuelo y tiene la motivación de seguir sus pasos y cumplir con su última voluntad.

En cuanto a su apariencia física, Ansel un niño con ropa de explorador y un mapa que le ayuda en su viaje por Deltaland. Tiene dos limitaciones principales:

* Vida: tiene 100 puntos de vida, que puede perder cuando es golpeado por un enemigo o recuperar con una poción de vida.
* Maná: tiene 100 puntos de maná, que gasta cuando realiza ataques especiales, es posible recuperar los puntos con una poción de maná.

### Amigos

Algunos de los personajes que encontremos por Deltaland serán amigables y podremos colaborar con ellos realizando las misiones o *quests* que nos manden. A cambio, recibiremos una recompensa.

### Enemigos

Los enemigos se pueden encontrar por toda la isla, pueden ser otros buscadores de tesoros o extrañas criaturas, al acabar con ellos recibiremos recompensas o *loot*.

# Diseño

## Entorno de desarrollo escogido

Como se ha especificado en la comparativa del punto anterior, el motor con el que se llevará a cabo el desarrollo de este videojuego será **Unity.** En concreto hemos comenzado a desarrollar con la **versión 2021.3.23f1** y posteriormente hemos actualizado a la **versión 2021.3.26f1** que corrige algunos errores del programa.

Este potente motor de desarrollo de videojuegos nos permite crear tanto juegos 2D como 3D, así como otro tipo de experiencias interactivas de una forma totalmente gratuita. Es por esto y por la gran cantidad de documentación que ofrece que es el motor favorito de los desarrolladores indie. Gracias a esto también podemos encontrar ayuda en foros y comunidades que nos facilitarán mucho el desarrollo del videojuego.

Unity nos ofrece una gran integración multiplataforma por lo que fácilmente podremos desarrollar para PC, Mac y muchas más plataformas con un mismo proyecto.

Logotipo

Descripción generada automáticamente

Figura 9. Logotipo de Unity

No solo es compatible con gran cantidad de plataformas, sino que también con diversas aplicaciones de modelado y animación por lo que no sufriremos por faltas de compatibilidad con los formatos más populares durante el proceso de desarrollo.

Podemos destacar algunos aspectos técnicos del motor como los siguientes:

* Nos permite desarrollar en C#, un lenguaje muy extendido y utilizado a día de hoy en numerosos ámbitos.
* Nos da la libertad para trabajar con el IDE que más nos guste. Por defecto utiliza MonoDevelop pero sencillamente podemos cambiarlo por VisualStudio como en este proyecto, o por Rider, Atom u otros disponibles en el mercado.
* Ofrece un extenso soporte para plugins, es decir, librerías de terceros que añaden nuevas funcionalidades y tecnologías al entorno de desarrollo.
* En primera instancia, se basa en escenas (scene) donde creamos el nivel del juego, puede contener cualquier cosa como objetos, personajes o incluso un menú.
* En segunda instancia, se basa en objetos de juego (gameobject) que basan la unidad principal de cada escena, ya que cualquier elemento que añadamos a la escena será un objeto y que podrá tener componentes asociados.
* Por último y en tercera instancia, se basa en componentes (component), lo que permite reutilizar y estructurar las diferentes entidades o propiedades del proyecto, como un script o una textura, y facilita su desarrollo y optimización. Se agrega a los objetos de una forma muy sencilla.

Podremos descargarlo desde su página web oficial a través del siguiente enlace:

<https://unity.com/es/download>

### Interfaz

Gráfico

Descripción generada automáticamente con confianza baja

Figura 10. Interfaz de Unity

* + - 1. **Menú de configuración**

En la parte superior encontramos un menú horizontal integrado en el sistema operativo que contiene todas las opciones del motor de videojuegos entre sus diferentes apartados (File, Edit, Assets, GameObject, Component, Jobs, Window y Help). Por ejemplo, en el menú File podremos guardar o cargar un nuevo proyecto, y en el GameObject crear cualquier tipo de objeto para el juego como sería un *tilemap* o un texto.

* + - 1. **Menú de navegación**

Debajo del anterior menú, encontramos algunos de los controles más usados de Unity como los botones de *play / pause* con los que podremos probar las implementaciones del juego ejecutándolo. También encontramos información relacionada con nuestra cuenta de usuario y algunos accesos rápidos como la búsqueda global o el historial de cambios.

* + - 1. **Barra de herramientas**

En este panel encontramos las principales herramientas del programa para trabajar en la escena, como son la de selección, mover, rotar, escala o punto de pivote.

* + - 1. **Vista y menú de escena**

En la pestaña “Scene” añadiremos todos los elementos del juego, los transformaremos e interactuaremos con ellos, es una especie de cámara externa al juego especialmente diseñada para crearlo. Sobre la escena encontramos el menú con las opciones necesarias para ajustar su visualización. También tenemos en la pestaña “Game” la cámara del juego, que al contrario que la anterior esta sí que es la que veremos cuando juguemos al mismo.

Por último, encontramos la pestaña “Animator” en la que programaremos las diferentes animaciones.

* + - 1. **Panel de jerarquía**

Es una lista organizada con todos los objetos que conforman la escena como los personajes, las decoraciones o los elementos de la interfaz. Funciona mediante una estructura de carpetas de padre e hijos, esto quiere decir que unos objetos pueden contener otros y beneficiarse de estas relaciones. Ejemplo: podemos asignar un arma (hijo) a nuestro personaje (padre) para que estos dos elementos se muevan conjuntamente en la escena.

* + - 1. **Menú de proyecto**

Este menú va más allá que el anterior, ya que podemos ver todos los archivos del proyecto y no solo los añadidos a la escena. Nos permitirá organizar la estructura del proyecto e importar recursos de todo tipo a la escena. Es ideal para esta tarea ya que nos da una vista previa de las imágenes, los audios o incluso los modelos 2D y 3D.

* + - 1. **Inspector**

Cuando seleccionamos un objeto en el menú de escena o de proyecto, nos muestra sus diferentes propiedades y componentes. Nos permite ajustar cosas como la posición, escala o colisiones de los elementos del juego y de esta forma personalizar su comportamiento.

* + - 1. **Tile Palette**

Este menú es una herramienta que nos ayuda a crear y usar piezas gráficas llamadas "tiles" para construir niveles en 2D. Estas piezas se pueden colocar en el juego para hacer el mundo y permite que este sea un proceso sencillo y rápido sin la necesidad de tener que escribir código. Es especialmente útil para videojuegos 2D como RPG o plataformas.

### Requerimientos de hardware

**Unity Hub**

Sistema operativo: Windows 7 (SP1+), 8, 10 y 11 64 bit, macOS 10.2 o superior, Ubuntu 16.04 o 18.04, CentOS 7

CPU: X64 | Tarjeta gráfica: compatible con DirectX 10, Metal

**Unity Editor (2021)**

Sistema operativo: Windows 7 (SP1+), 10 y 11 64 bit, macOS High Sierra 10.13 o superior, Ubuntu 18.04 o 20.4, CentOS 7

CPU: X64 | Tarjeta gráfica: compatible con DX10, DX11 y DX12, Metal

## Assets

Los *assets* son los recursos que utiliza el videojuego en la etapa de su creación, es decir, en su desarrollo dentro del motor de videojuegos.

Hay numerosas páginas externas que nos permiten descargar de formas gratuita gran cantidad de *assets*, pero en Unity contamos con la Asset Store, una tienda en la que encontramos todo tipo de recursos para nuestro proyecto, tanto de forma gratuita como de pago.

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Figura 11. Logotipo de la Unity Asset Store

Ya que para el desarrollo de un videojuego *indie* como este no disponemos del tiempo necesario para crear desde cero personajes y otros elementos, ha sido necesario descargar una serie de recursos:

**Principales assets de texturas y animaciones**: [Aprende a crear un videojuego RPG en 2D con Unity | Udemy](https://www.udemy.com/course/aprende-a-crear-un-videojuego-rpg-en-2d-con-unity-2021/)

## Arquitectura del proyecto

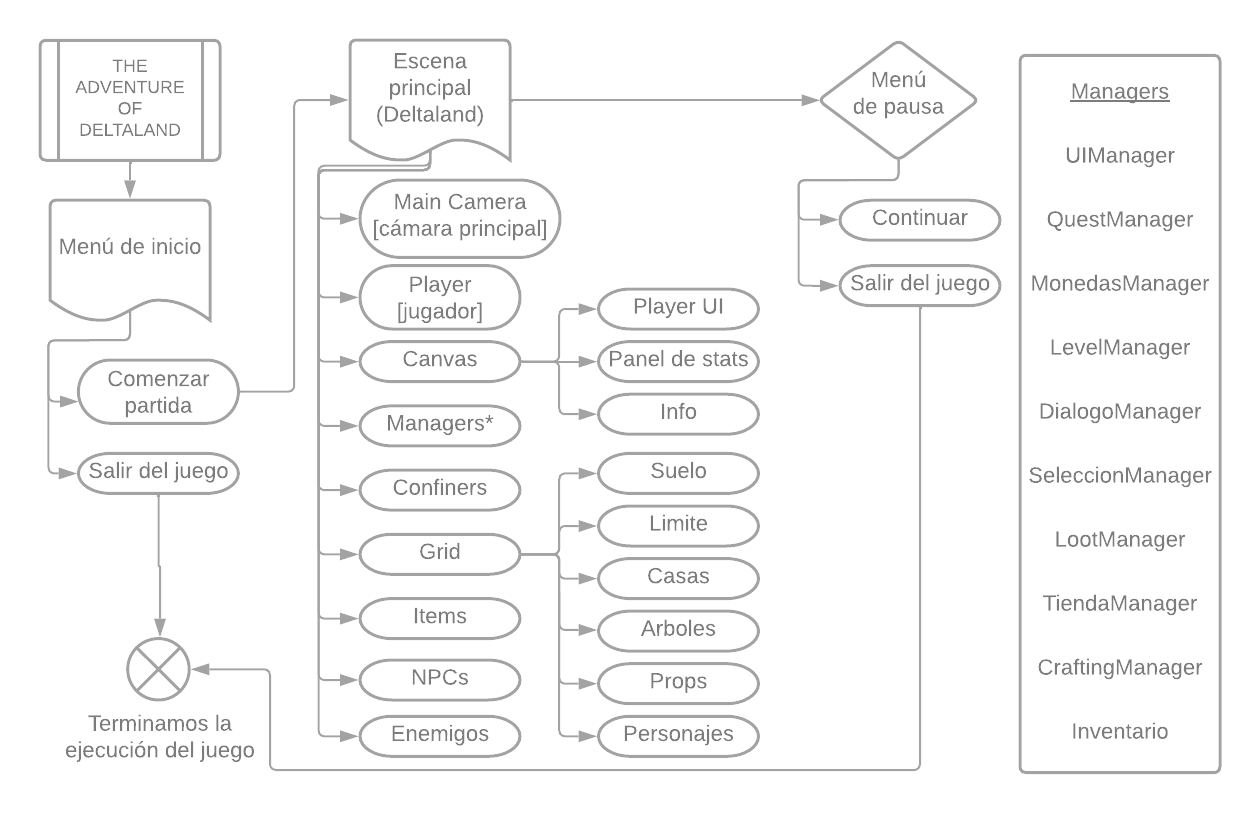


Tabla 4. Arquitectura del proyecto

## Proceso de desarrollo

Vamos a explicar, paso por paso, como ha sido el proceso de desarrollo del proyecto “The Adventure of Deltaland”.

### Creación del proyecto

Tras haber instalado el motor de videojuegos Unity y los programas requeridos, el primero de los pasos es crear un nuevo proyecto desde Unity Hub, y con la versión de Unity Editor que deseemos.

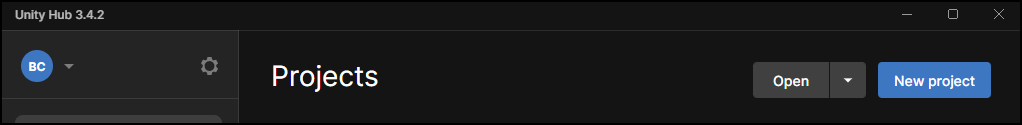


Figura 12. Nuevo Proyecto de Unity

Es muy importante saber la diferencia entre un proyecto 2D y otro 2D (URP). El segundo utiliza Universal Render Pipeline, un renderizador de alto rendimiento que nos proporciona características adicionales como iluminación global en tiempo real, reflejos u otros efectos visuales avanzados. Son características que requieren de un mayor conocimiento técnico y esfuerzo de optimización por lo que no es ideal para este proyecto, en el que buscamos crear nuestro primer videojuego y aprender las funciones básicas que nos ofrece este potente motor de desarrollo.

Por tanto, elegimos la versión adecuada y el renderizador estándar 2D.

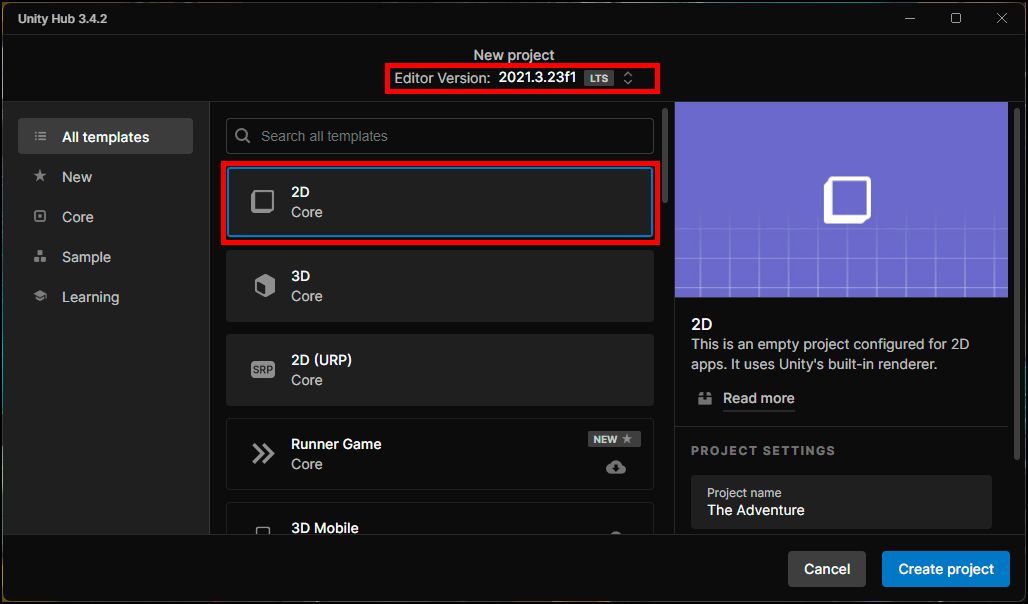


Figura 13. Renderizador

### Personaje principal

El primer paso en el desarrollo de nuestro videojuego es el diseño del personaje principal y sus funciones básicas de movimiento. Para ello debemos desarrollar el sistema para que el sprite del personaje pueda moverse por el espacio mediante unas teclas asignadas.

Después se deben aplicar las animaciones y para ello cada una debe activarse en su momento preciso, para esto se hace uso de la herramienta Animator que nos ofrece Unity, y creamos un árbol de animaciones para cada una de ellas.

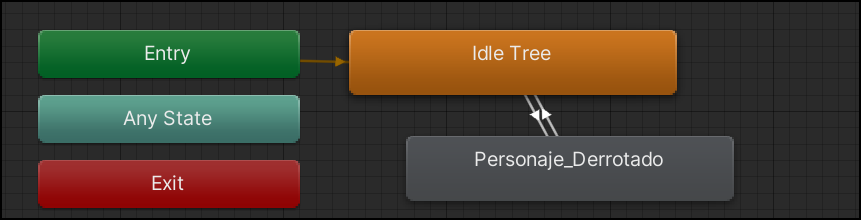


Figura 14. Animaciones

Una captura de pantalla de un celular

Descripción generada automáticamente

Figura 15. Blend Tree

### Interfaz

La interfaz del juego permite al jugador recibir información relevante sobre el estado del personaje que controla, así como interactuar con el propio juego para realizar diferentes acciones. En este caso hemos diseñado una interfaz es muy sencilla e intuitiva y la que veremos en todo momento en pantalla cuenta con cuatro elementos principales:

* **Barra de salud:** muestra el estado del personaje, indicando cuánta vida le queda. Al llegar a 0 el personaje muere.
* **Barra de maná:** similar a la barra de vida, indica la cantidad de maná de la que dispone el personaje para realizar ataques. Cada uno de estos consumirá una cantidad determinada de maná hasta agotarse. Para recuperar maná, basta con esperar unos minutos ya que se recupera cierta cantidad por segundo.
* **Barra de experiencia:** indica los puntos de experiencia del nivel actual y al llenarla nos permite subir de nivel. Cuando esto ocurre, la barra se vacía y el siguiente nivel requiere de una mayor cantidad de experiencia.
* **Indicador de nivel:** simplemente nos muestra en qué nivel se encuentra nuestro personaje y dependiendo de ello tendremos unas mejores stats que nos ayudarán en combate.

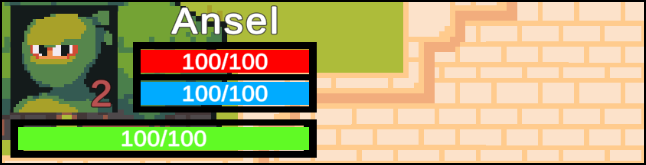


Figura 16. Interfaz del juego

Por otro lado, contamos con otro tipo de interfaces compuestos por diferentes campos de texto y botones, estos son:

* **Panel de stats:** este menú nos muestra las diferentes estadísticas del personaje. También cuenta con el apartado de atributos que muestra los puntos de atributo obtenidos al subir de nivel y los propios atributos que podemos subir para mejorar nuestras habilidades.



Figura 17. Panel de Stats

* **Inventario:** este panel permite al jugador gestionar y organizar los objetos y recursos obtenidos. Es un elemento esencial para el comercio y la progresión del juego. Imagen que contiene Diagrama

  Descripción generada automáticamente

Figura 18. Panel de Inventario

* **Quests activas**: este panel muestra las misiones que hemos aceptado a los NPCs y en este momento están activas. Interfaz de usuario gráfica

  Descripción generada automáticamente

Figura 19. Panel de Quests activas

Tanto el comercio como el crafting hacen uso de variaciones de esta interfaz para crear un aspecto visual más coherente.

* **Diálogos:** la interfaz de los diálogos nos permite interaccionar con los NPCs y de este modo nutrirnos de la historia y el trasfondo que nos quiere transmitir el videojuego.

### Mapa

La creación del mapa es uno de los principales procesos en la creación del videojuego, es el entorno en el que se desarrollará esta aventura y se realiza mediante una serie de pasos bien definidos:

* **Diseño del mundo:** en primer lugar debemos definir los principales conceptos del mundo, dejando claro cuál va a ser el objetivo de cada zona del mapa y en qué lugar va a estar ubicada.

Mapa

Descripción generada automáticamente

Figura 20. Mapa de Deltaland

Ya que en esta aventura nos encontramos en la isla de Deltaland, la isla ha sido diseñada especialmente pensando en este concepto, es por ello que tiene una forma similar a la letra griega Delta (minúscula) [δ].

* **Tiles y tilemaps:** el siguiente paso es la creación de estas pequeñas imágenes (tiles) que nos sirven como unidad básica de construcción del mapa. Con los tilemaps podemos colocar los tiles de forma más eficiente como aleatorizar la textura de la hierba o colocar esquinas automáticamente.



Figura 21. Tiling Rules

* **Colisiones y físicas:** las colisiones se utilizan para que los personajes y objetos interactúen correctamente con el mapa. Definimos áreas sólidas para evitar que el personaje atraviese objetos o se salga fuera del mapa.
* **Decoración y detalles:** son elementos imprescindibles para mejorar la experiencia visual del juego. Algunos de estos elementos añadidos son árboles, rocas u otros objetos.

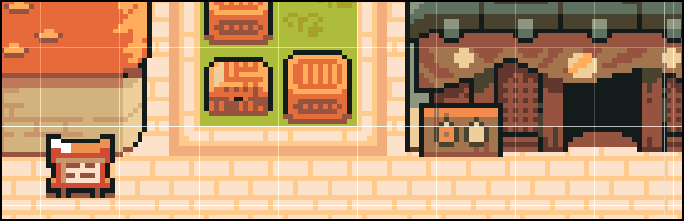


Figura 22. Decoraciones

### NPCs y diálogos

En esta etapa, nos enfocamos a dar vida a nuestros personajes o comúnmente llamados NPCs, esto implica considerar diferentes aspectos:

* **Definimos la apariencia visual de cada NPC:** es importante diseñar tanto sus proporciones como paleta de colores para transmitir la esencia y estilo del videojuego.

Por ejemplo, los enemigos tienden a utilizar colores cercanos al rojo y al negro, ya que están fuertemente relacionados con el concepto de “mal”. Con ello, nos transmiten el peligro, la violencia y la ira de los enemigos a los que nos enfrentamos. Por otra parte, el color verde que lleva el personaje principal nos transmite una sensación de calma y seguridad.

Como vemos, es muy importante tener en cuenta la psicología del color entre otros aspectos a la hora de diseñar los personajes que componen el videojuego.

* **Definimos la narrativa de cada NPC:** ya que es un videojuego con una gran historia detrás, es importante que nuestros personajes tengan un trasfondo personalidades únicas, lo que nos lleva a una experiencia mucho más inmersiva y enriquecedora para que el jugador conecte emocionalmente con el videojuego.

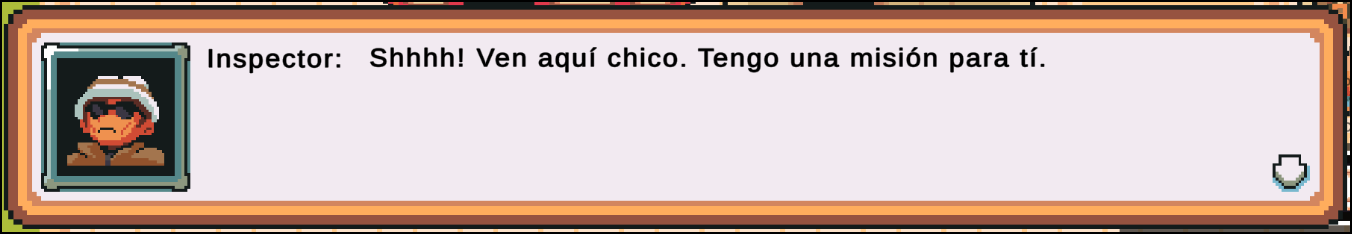
****

Figura 23. Diálogos

### Lootdrop

Este concepto consiste en la generación aleatoria de objetos y recompensas para el jugador por haber derrotado enemigos o haber completado desafíos. Nos agrega emoción al videojuego y mejora enormemente la experiencia ya que con algunos de estos objetos tenemos la posibilidad de mejorar a nuestro personaje y derrotar a los enemigos de forma más sencilla.

Hemos diseñado una serie de recompensas que harán que nuestro viaje por la isla de Deltaland sea mucho más interesante.

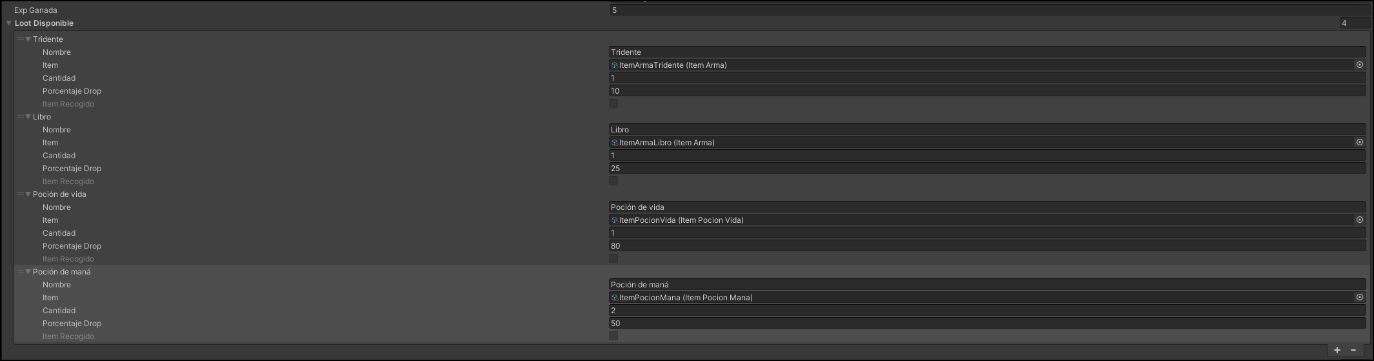
****

Figura 24. Ejemplo de lootdrop de un enemigo

### Misiones

Los diferentes NPCs que encontramos por el mapa pueden encomendarnos una serie de misiones o “quests”, en las que normalmente realizamos tareas sencillas como acabar con una cierta cantidad de enemigos.



Figura 25. Panel de Quests del Inspector

### Combate y puntos de experiencia (XP)

Aunque numerosos RPG se basan en el sistema de combate por turnos, en este proyecto hemos decidido crear un sistema más dinámico conocido como combate en tiempo real, en el que la acción se desarrolla de forma continua y sin interrupciones. Es importante mejorar las stats del personaje y decidir en cada momento si es mejor utilizar un ataque cuerpo a cuerpo o uno a distancia.

Al acabar con los enemigos obtendremos puntos de experiencia (XP) con los que subiremos de nivel a nuestro personaje y esto nos otorgará la posibilidad de gastar puntos de atributo para mejorar nuestros stats, así como de obtener lootdrop.

****

Figura 26. Combate

### Comercio y crafting

Tanto el comercio como el crafting (creación de objetos mediante otros materiales) son elementos muy importantes en un videojuego RPG como el que estamos desarrollando, ya que brindan al jugador la posibilidad de obtener recursos valiosos y con ello avanzar en el juego de una forma mucho más sencilla. Algunos comerciantes pueden permitirnos adquirir objetos únicos y diferentes materiales.

****

Figura 27. Panel de la tienda

****

Figura 28. Panel de crafting

# Implementación

## Requisitos de instalación

Para asegurar el funcionamiento adecuado del juego, es importante contar con los siguientes requisitos del sistema:

* **Sistema operativo:** El juego es compatible con los sistemas operativos Windows 10 / 11, macOS y Linux.
* **Procesador:** Se recomienda un procesador de al menos 2.5 GHz para un rendimiento óptimo.
* **Memoria RAM:** Se recomienda tener al menos 4 GB de RAM para una experiencia de juego fluida.
* **Tarjeta gráfica:** Es necesario contar con una tarjeta gráfica compatible con DirectX 11 o superior.
* **Espacio en disco:** Se requerirá un mínimo de 5 GB de espacio libre en disco para almacenar el juego y sus archivos relacionados.

No se requiere ningún conocimiento específico para la instalación del videojuego, ya que se trata únicamente de un archivo ejecutable.

## Instrucciones de instalación

El juego se puede obtener y ejecutar siguiendo los siguientes pasos:

* Accede a la página de GitHub del proyecto [[bernatrodriguez/The-Adventure-of-Deltaland: My first RPG game using Unity 2D - UOC TFG (github.com)](https://github.com/bernatrodriguez/The-Adventure-of-Deltaland)].
* Dentro de “Builds”, descarga la carpeta “Gold Master (final)” que contendrá un archivo ejecutable (".exe") y el resto de los recursos del juego. También puedes encontrar una descarga directa en la descripción.
* Una vez finalizada la descarga, abre la carpeta donde se haya guardado el archivo ejecutable.
* Haz doble clic en el archivo ejecutable del juego para iniciarlo.
* Disfruta del juego y explora la isla de Deltaland.

Nota: es posible que se requieran permisos de administrador en el sistema operativo al abrir el archivo ejecutable.

# Demostración

## Controles del juego

Los controles del juego han sido diseñados de la manera más intuitiva y sencilla posible para asegurar la mejor experiencia de juego.



Figura 29. Controles del juego

Los controles principales son:

* **Menús y paneles:** Usa el botón izquierdo del ratón para interactuar con los paneles del juego, como seleccionar un *item* o añadir puntos de atributo. Pulsa la tecla [M] para mover *items* dentro del inventario.
* **Movimiento del personaje**: Utiliza las teclas [W], [S], [A] y [D] o las teclas de dirección [↑], [↓], [←] y [→] para mover al personaje hacia adelante, atrás, izquierda o derecha.
* **Interacción con NPCs:** Utiliza la tecla de acción [E] para interactuar y hablar con los personajes del juego. Usa la tecla [Espacio] para pasar los diálogos.
* Ataque y habilidades: Utiliza el ratón para apuntar a los enemigos y [Espacio] para lanzar ataques básicos.
* **Reaparición:** Si mueres, utiliza la tecla [R] para reaparecer en el punto de aparición.
* **Menú y opciones:** Puedes acceder al menú de pausa utilizando la tecla [Esc].

## Tests

Durante el desarrollo del proyecto, se han llevado a cabo diferentes pruebas para garantizar la funcionalidad y el buen rendimiento del juego. A continuación, se explican los tests realizados:

* **Tests de usabilidad:** Se han realizado pruebas de usabilidad para valorar la facilidad de uso de la interfaz del videojuego. Se ha determinado que, al ser una interfaz tan simple, constatada por los parámetros del jugador en tiempo real y tres paneles, es completamente usable para la mayoría de los jugadores.
* **Tests de seguridad:** Se han hecho pruebas de seguridad y se han corregido errores para evitar mecánicas abusivas y no deseadas en el juego, como superposiciones.

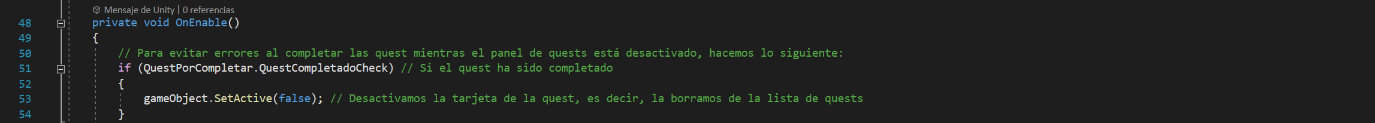
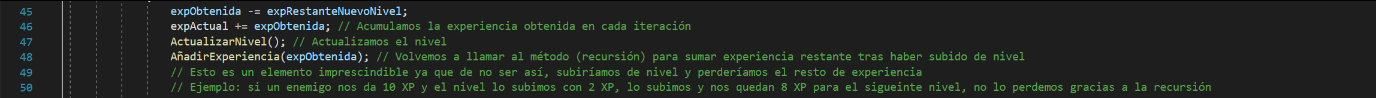
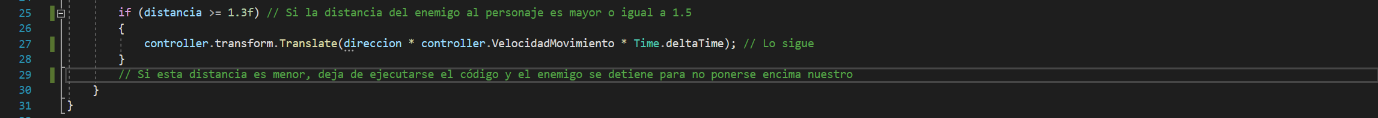
****

Figura 30. Tests de seguridad

* **Pruebas de rendimiento:** Se ha probado el rendimiento del videojuego en diferentes máquinas y sistemas operativos para evaluar su estabilidad general, teniendo en cuenta la carga de recursos que supone y su optimización.

|  |  |  |  |
| --- | --- | --- | --- |
| **The Adventure of Deltaland**  *Valores de rendimiento* | | | |
| CPU | GPU | Memoria | Disco |
| 0 - 5 % | 0 - 10 % | 100 - 200 MB | 0 – 10 MB/s |

Tabla 5. Valores de rendimiento

## Guía de usuario

Aunque es cierto que un videojuego debe explorarse y disfrutar libremente, a continuación, se muestra un ejemplo práctico que puede guiar al usuario dentro del videojuego:

**1. Completar una misión:**

Paso 1: Explora el mapa y habla con el inspector para ver las misiones que tiene para ti.

Paso 2: Selecciona una misión de la lista, lee su descripción y objetivos.

Paso 3: Interactúa con los NPCs para obtener información que te puede ser de ayuda.

Paso 4: Ves al lugar que consideres oportuno para completar la misión.

Paso 5: Completa todos los objetivos de la misión.

Paso 6: Cuando lo hagas, recibirás una recompensa, oro y experiencia.

**2. Mejorar equipo y habilidades:**

Paso 1: Accede al inventario del personaje para ver los objetos disponibles.

Paso 2: Selecciona un objeto que deseas equipar o tomar, como una nueva arma o poción.

Paso 3: Usa el botón de acción adecuado para cada tipo de objeto.

Paso 4: Accede al menú de habilidades para asignar puntos de atributo a las habilidades deseadas.

**3. Creación y compra de objetos**

Paso 1. Explora el mundo del juego y recolecta recursos.

Paso 2. Encuentra un herrero para crear objetos a partir de recursos o un comerciante.

**4. Exploración del mundo:**

Paso 1: Utiliza los controles de movimiento (teclas de dirección o teclas WSAD) para desplazarte por el mapa.

Paso 2: Explora diferentes áreas, interactúa con el entorno y descubre secretos ocultos.

# Conclusiones y líneas de futuro

## Conclusiones

Durante la realización de mi trabajo sobre el diseño y desarrollo del videojuego “The Adventure of Deltaland”, he aprendido muchas lecciones. Ha sido una experiencia muy enriquecedora y que me ha permitido adquirir nuevas habilidades en el campo del desarrollo de videojuegos. He mejorado exponencialmente mi nivel de programación en C# y el manejo de Unity, uno de los motores gráficos más utilizados en la industria.

Si pienso en los objetivos que me planteé al inicio del proyecto, puedo decir que en su mayoría los he cumplido, pero debido a las limitaciones de tiempo y recursos, puede que no haya alcanzado algunos objetivos específicos como una historia lo suficientemente profunda y extensa. El diseño y desarrollo de un juego RPG en Unity es sin duda un gran desafío que requiere de horas de programación y diseño de recursos.

Al analizar la planificación y metodología del proyecto, puedo decir que me he alejado bastante de la planificación establecida inicialmente, ya que he tenido que realizar numerosos ajustes y adaptaciones según las necesidades de cada momento, cambios que han sido necesarios para el éxito del proyecto.

También he tenido numerosos problemas con el sistema de combate que me han hecho perder una gran cantidad de tiempo investigando para poder resolverlos

En resumen, el proceso de diseño y desarrollo del videojuego ha sido una experiencia muy valiosa para mí, y aunque no he logrado alcanzar todos los objetivos en su totalidad, he tenido resultados muy significativos a pesar de todos los problemas en el desarrollo. Este proyecto me ha permitido conocer mis límites y sentar las bases para futuros proyectos mucho más ambiciosos.

## Líneas de futuro

Ahora que ha terminado el diseño y desarrollo del videojuego, es emocionante pensar en sus posibles direcciones de cara a futuro. Estas son algunas ideas para seguir mejorando la experiencia del juego:

* **Mejor historia:** Es el punto más débil del videojuego, la absoluta prioridad para una versión futura es mejorar enormemente la historia, añadiendo un trasfondo a cada personaje y numerosos *easter eggs*.
* **Más contenido:** Creación de nuevas zonas jugables con más combates y misiones para que los jugadores disfruten de más horas de diversión. También una mayor cantidad de *NPCs* con posibles nuevas interacciones.
* **Mejoras en la jugabilidad:** Se pueden pulir las mecánicas de combate, exploración y progresión del personaje para tener una experiencia mucho más fluida.
* **Más plataformas:** Sería ideal que una mayor cantidad de jugadores puedan probar el juego, para ello es imprescindible adaptarlo y publicarlo en otras plataformas como iOS y Android. También es interesante publicarlo en Steam, ya que además de estar contemplado en el presupuesto, es la mejor forma de hacer llegar este tipo de videojuegos al público general.
* **Jugar en equipo:** Agregar un modo multijugador para que varios jugadores puedan unirse y jugar juntos. Vivir la aventura y explorar la isla de forma cooperativa, librar emocionantes batallas con amigos e incluso añadir nuevas funciones exclusivas para este modo.

Estas son solo algunas ideas para el futuro del videojuego “The Adventure of Deltaland”. Su principal objetivo es que siga en constante actualización y que todo el mundo pueda tener la oportunidad de probarlo.

Glosario

* ***Asset:*** Recurso gráfico, sonoro o de programación utilizado en el desarrollo de un videojuego.
* **Diagrama de Gantt:** Herramienta visual que muestra la planificación y programación de tareas en un proyecto, incluyendo sus fechas de inicio y finalización.
* ***Easter egg:*** Elemento oculto dentro de un videojuego, como una referencia o mensaje secreto, y que los jugadores pueden descubrir.
* **GUI (Interfaz gráfica de usuario):** Componente visual que permite a los jugadores interactuar con el juego utilizando elementos gráficos como botones y paneles.
* **IDE (Entorno de desarrollo integrado):** Software que proporciona herramientas y funcionalidades para facilitar el desarrollo y programación de aplicaciones, incluyendo videojuegos.
* **Loot:** Recompensa obtenida por los jugadores al derrotar enemigos o completar misiones en un videojuego.
* **Maná**: Energía utilizada en algunos juegos de rol para activar habilidades mágicas o poderes especiales.
* **Motor de videojuegos:** Software que proporciona las herramientas y librerías necesarias para el desarrollo de un videojuego, como la gestión de gráficos, físicas y sonidos.
* **NPC (Personaje no jugador):** Personaje controlado por la inteligencia artificial del videojuego, no por el jugador.
* ***Open source:*** Software cuyo código fuente es accesible y puede ser modificado y distribuido libremente por la comunidad.
* ***Plugins:*** Complementos que se pueden añadir a un motor de videojuegos para agregar funcionalidades adicionales o características específicas.
* ***Quest:*** Tarea o misión que los jugadores deben completar dentro de un videojuego, con un objetivo específico y recompensa.
* **RPG (Juego de rol):** Género de videojuegos donde los jugadores asumen el papel de un personaje y toman decisiones que afectan su desarrollo y el desarrollo de la historia.
* ***Script:*** Código de programación que define acciones o comportamientos específicos en un videojuego. Los scripts controlan la lógica del juego y la interacción con los jugadores.

Bibliografía

* Unity (2023). *Unity Solutions for 2D*. Unity. <https://unity.com/solutions/2d>
* Gianny Dantas (2021). *Aprende a crear un videojuego RPG en 2D con Unity*. Udemy. <https://www.udemy.com/course/aprende-a-crear-un-videojuego-rpg-en-2d-con-unity-2021/>
* Redacción de GamersRD (2023). *Con estos motores gráficos podrás iniciar tu carrera como desarrollador de videojuegos*. GamersRD. <https://gamersrd.com/con-estos-motores-graficos-podras-iniciar-tu-carrera-como-desarrollador-de-videojuegos/>
* juicy\_fish (2023). *Teclado realista | Vector gratis*. Freepik. <https://www.freepik.es/vector-gratis/teclado-realista_35202474.htm>