

Shaky Town

Miquel Bellet Coll

Máster Universitario en Diseño y Programación de Videojuegos
Trabajo Final de Máster

Jordi Duch Gavalda
Helio Tejedor Navarro
Joan Arnedo Moreno

06/2021



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Shaky Town</i>
Nombre del autor:	<i>Miquel Bellet Coll</i>
Nombre del consultor/a:	<i>Jordi Duch Gavalda</i>
Nombre del PRA:	<i>Joan Arnedo Moreno</i>
Fecha de entrega (mm/aaaa):	<i>06/2021</i>
Titulación:	<i>Máster Universitario en Diseño y Programación de Videojuegos</i>
Área del Trabajo Final:	<i>Trabajo Final de Máster</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave:	<i>Videojuego, Unity, 2D</i>
<p>Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i></p>	
<p>Este proyecto se basa en la creación de un videojuego con el motor de desarrollo Unity versión 2020.4f1. Se ha creado un juego en 2D de acción-aventura para un solo jugador donde se tendrá que ir explorando los distintos lugares del mapa, luchando contra monstruos peligrosos e ir avanzando en la historia con ayuda de las notas que se van encontrando.</p> <p>Se ha decidido realizar este proyecto como trabajo final de máster ya que durante el curso se han hecho pequeños videojuegos de diferentes formas y se ha considerado que un buen proyecto final sería juntar los distintos conocimientos obtenidos en un solo proyecto.</p> <p>Empezando por el planteamiento, se planificaron todas las tareas generales para acabar el proyecto a tiempo y se utilizó un Diagrama de Gantt para ver el transcurso del tiempo con las tareas planificadas. Para el seguimiento de los objetivos, se utilizó la metodología Scrum para completar las tareas con un tiempo corto y revisar el proyecto cada cierto tiempo para actualizar esas tareas y seguir desarrollado el videojuego. Para algunos aspectos gráficos se han utilizado librerías de recursos con licencia de comercialización para utilizar-las en el proyecto y en un futuro poder distribuir-lo.</p> <p>El resultado del proyecto ha sido un videojuego completo y funcional con una historia, mecánicas y gráficos bien implementados. Hay algunos aspectos del juego que se podrían mejorar para hacerlo más intuitivo y atractivo para el usuario, pero se han conseguido la mayoría de objetivos planteados al comienzo del proyecto.</p>	

Abstract (in English, 250 words or less):

This project is based on the creation of a **video game** with the **Unity** game engine version 2020.4f1. A **2D action-adventure** game has been developed for a single player where you have to explore, fight against monsters and progress in the story with the help of the notes that are found.

It has been decided to make this project as a final master work because during the course small video games have been made in different ways and I have considered that a good final project would be to **bring together the different knowledge** obtained in a single project.

Starting with the approach, all the general tasks were planned to finish the project on time and a **Gantt Chart** was used to see the course of time with the planned tasks. For the follow-up of the tasks, the **Scrum** methodology was used to complete the tasks in a short time and review the project from time to time to update the tasks and continue developing the video game. For some graphical aspects we have used **resource packages** with commercial license to use them in the project and in the future to be able to distribute it.

The result of the project has been a complete and functional videogame with a well implemented story, mechanics and graphics. There are some aspects of the game that could be improved to make it more intuitive or attractive to the user, but **the objectives** set at the beginning of the project **have been achieved**.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	1
1.3 Enfoque y método seguido.....	2
1.4 Planificación del Trabajo	2
1.5 Breve resumen de productos obtenidos	3
1.6 Breve descripción de los otros capítulos de la memoria.....	3
2. Estado del arte	4
2.1 Decisión de tecnología usada	4
2.2 Estudio de alternativas	4
2.3 Dispositivo final	5
3. Definición del juego	6
3.1 Breve descripción del juego	6
3.2 Subgénero y referencias a videojuegos existentes	6
3.3 Tipo de interacción juego-jugador	6
3.4 Ambientación.....	7
3.5 Historia	8
3.6 Definición de personajes	9
3.7 Tipos de enemigos	10
3.8 Objetivos planteados al jugador	10
3.9 Concept Art	10
4. Diseño técnico.....	12
4.1 Entorno y requerimientos	12
4.2 Herramientas empleadas y recursos	12
4.3 Arquitectura del juego.....	15
4.4 Funcionamiento de las Inteligencias Artificiales	17
5. Diseño de niveles	18
5.1 Tutorial	18
5.2 Pueblo	18
5.3 Bosque	19
5.4 Playa	20
5.5 Bosque Oscuro.....	20
5.6 Cueva Secreta.....	21
6. Manual de usuario.....	22
6.1 Instalación	22
6.2 Controles	23
7. Conclusiones.....	24
7.1 Conclusiones del trabajo	24
7.2 Reflexión sobre los objetivos planteados	25
7.3 Seguimiento de la planificación y metodología.....	25
7.4 Líneas de trabajo futuro.....	26
8. Glosario	27
9. Bibliografía	29
10. Anexos	30

Lista de figuras

<i>Figura 1. Diagrama de Gantt.....</i>	<i>2</i>
<i>Figura 2. Unity vs Unreal vs Godot.....</i>	<i>5</i>
<i>Figura 3. Ambientación del juego</i>	<i>7</i>
<i>Figura 4. Personajes del juego.....</i>	<i>9</i>
<i>Figura 5. Casa del Herrero.....</i>	<i>11</i>
<i>Figura 6. Carpetas dentro de "Assets"</i>	<i>15</i>
<i>Figura 7. Archivos de la carpeta de "Resources"</i>	<i>16</i>
<i>Figura 8. Archivo de configuración del murciélago.....</i>	<i>17</i>
<i>Figura 9. Mapa del tutorial.....</i>	<i>18</i>
<i>Figura 10. Nivel 1 - Pueblo principal.....</i>	<i>19</i>
<i>Figura 11. Nivel 2 - Bosque</i>	<i>19</i>
<i>Figura 12. Nivel 3 - Playa</i>	<i>20</i>
<i>Figura 13. Nivel 4 - Bosque Oscuro</i>	<i>21</i>
<i>Figura 14. Nivel 5 - Cueva secreta</i>	<i>21</i>
<i>Figura 15. Instalador de Windows.....</i>	<i>23</i>

1. Introducción

1.1 Contexto y justificación del Trabajo

Para este trabajo final se quiso realizar un proyecto **juntando** muchos de los **conocimientos obtenidos** hasta el momento y utilizar los ámbitos aprendidos en el máster que convinieran para llegar a un resultado final **completo**.

La industria de los videojuegos está creciendo exponencialmente y cada año se venden millones de juegos en todo el mundo y se crean nuevas compañías desarrolladoras para poder ofrecer tanto contenido a los consumidores. [1]

Desde el principio del planteamiento de este proyecto, se buscó terminar con un **videojuego completo** con buenas mecánicas, historia, objetivos, gráficos acabados, buen diseño de niveles, curva de dificultad y diversión para que el jugador pudiera disfrutarlo.

1.2 Objetivos del Trabajo

- Terminar un **videojuego** con Unity **completo y jugable**.
- Que el juego sea en **2D** y con **mecánicas trabajadas**.
- Añadir **gráficos decentes** para la comodidad y disfrute del jugador.
- Que haya **varios niveles** con **distintos enemigos**.
- El usuario ha de disponer de una **barra de objetos** que lleva encima.
- Los objetos se pueden **tirar al suelo, recoger y utilizar**.
- Que se puedan **guardar objetos** en un **cofre**.
- Permitir al jugador **guardar la partida**, salir y que cuando vuelva a iniciar el juego siga en el **mismo punto**.
- Obtener **recursos** para poder **comprar** nuevas **armas y pociones**.
- Un sistema de **venta** para obtener oro a **cambio de objetos**.
- Crear un sistema de **combate dinámico**.
- Plantear una **historia y objetivos interesantes**.
- Que se pueda **seguir jugando** e investigando una vez **terminada** la historia.
- Permitir jugar una partida **desde cero** si se ha completado la historia.
- Poner distintos **niveles de dificultad** para hacer el juego más **accesible** a distintos tipos de usuarios.
- **Equilibrar** el coste, obtención y utilización de los recursos y objetos.
- Equilibrar la **dificultad** del juego con las habilidades de los enemigos.
- Añadir un apartado de **configuración** para poder cambiar el volumen de la **música** y los **efectos** sonoros y el nivel de **dificultad** del juego.

1.3 Enfoque y método seguido

Al plantearse como desarrollar el producto, se decidió crear un pequeño **GDD** (Game Design Document) donde se exponían todos los **objetivos** y **características** importantes que se querían implementar al juego final. Desde un principio, se tenía claro que la parte de **recursos gráficos y sonoros** no se crearía desde cero, sino que se descargarían **paquetes de recursos** para agilizar el proyecto y poder realizar un producto final más completo. Todos los recursos utilizados tienen una licencia de **comercialización libre**, la mayoría son gratuitos y alguno es de pago.

Para realizar las tareas planteadas, se utilizó la metodología **Scrum** para organizar el tiempo disponible y las tareas.

Se basa en crear unas tareas e ir realizándolas en un **periodo corto** de tiempo. Se van creando de nuevas, actualizando o completando tareas en función del **desarrollo del proyecto**.

1.4 Planificación del Trabajo

Teniendo claro los aspectos que se querían implementar, se utilizó un **Diagrama de Gantt** para planificar el tiempo que había disponible para finalizar el proyecto con los **puntos planificados**.

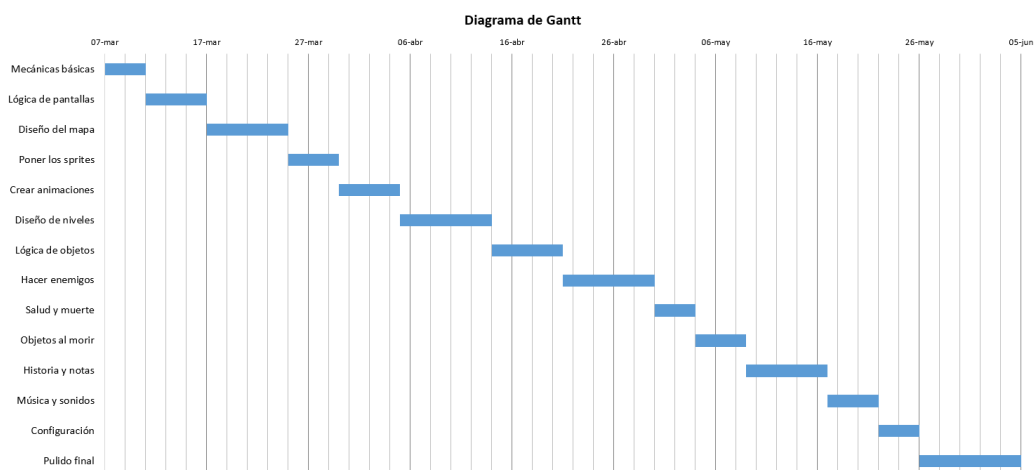


Figura 1. Diagrama de Gantt

Para la entrega de la PEC2, se tenían realizadas más o menos las 6 primeras tareas generales. Ya se habían programado las **mecánicas** básicas del personaje, tanto el movimiento como habilidades o combate con **enemigos**. También la organización de las pantallas y los niveles en el mundo planteado, al igual que los cofres. Ya se habían escogido todos los paquetes de recursos gráficos que se utilizarían para el proyecto y se habían implementado los básicos para comprobar que funcionaban bien.

Luego, para la PEC3, se aplicaron la mayoría de tareas restantes, como el **diseño y programación** del resto de **enemigos**, la salud del personaje y su reaparición en caso de muerte o los recursos obtenidos al acabar con enemigos. La parte de la **historia** se creó durante esta fase junto con las notas que se recogen en el suelo o la lógica y diálogos de la **escena final**.

Las tareas que se realizaron al final del proyecto fueron la implementación de **música** por nivel y los **efectos sonoros** del entorno, y crear un apartado de **configuración** donde se pudieran cambiar aspectos del juego como el volumen de la música y efectos sonoros y el nivel de **dificultad del juego**.

1.5 Breve resumen de productos obtenidos

El producto final obtenido se trata de un videojuego completo y funcional. Se puede jugar tanto en dispositivos **Windows** como en **IOS** con ejecutables fáciles de usar.

Se han **cumplido** la mayoría de los **objetivos planteados** en el inicio del proyecto y se han podido poner en práctica muchos **aspectos aprendidos y trabajados** a lo largo del curso. Aunque se pueden mejorar muchos puntos del producto final.

1.6 Breve descripción de los otros capítulos de la memoria

- **Estado del arte:** Revisión de la tecnología utilizada, plataformas de desarrollo y breve explicación.
- **Definición del juego:** Conceptualización e idea básica del juego.
- **Diseño técnico:** Entorno final elegido, requerimientos técnicos, herramientas y recursos empleados, arquitectura del juego y explicación del funcionamiento de la Inteligencia Artificial de los enemigos.
- **Diseño de niveles:** Explicar los mapas del juego y los criterios de su diseño.
- **Manual de Usuario:** Instrucciones de instalación y del juego.
- **Conclusiones:** Descripción de las conclusiones del trabajo y reflexión crítica.
- **Glosario:** Definición de los términos y acrónimos más utilizados.
- **Bibliografía:** Lista numerada de referencias bibliográficas utilizadas.
- **Anexos:** Listado de apartados demasiados extensos para añadir en el trabajo.

2. Estado del arte

2.1 Decisión de tecnología usada

El proyecto se trata de un videojuego de acción-aventura en 2D con gráficos al estilo “*pixelart*” de resolución 16x16. Estas características **influyeron** en la **decisión** de la tecnología que se usaría para el proyecto, pero lo que fue más importante, fue que todo el curso e incluso **proyectos anteriores** que se habían realizado, se habían hecho con el motor **Unity**.

Pero, aunque no se hubiese tenido ya experiencia con este motor, las características nombradas anteriormente **decantaban significativamente** la balanza para Unity.

2.2 Estudio de alternativas

Los “*Game Engines*” o **motores de desarrollo** de videojuegos más populares y utilizados del momento son Unity, Unreal Engine y Godot. [2]

- **Unity** es usado por pequeñas compañías o desarrolladores **indies** para hacer proyectos relativamente **pequeños**. Tiene una fuerte ayuda a proyectos de 2D y 3D y para videojuegos para dispositivos móviles.
- **Unreal Engine** en cambio, es usado por las **grandes compañías** para desarrollar **videojuegos triple A**. Está hecho para ser usado por grandes grupos de desarrolladores y está muy enfocado en las **capacidades gráficas** de simulación de realidad. No es el más indicado para ser usado para un proyecto indie 2D con “*pixelart*”.
- **Godot** no tiene tantos recursos ni una comunidad tan grande como las dos anteriores, aunque permite la **modificación del engine** y la **libre distribución** de sus proyectos. Tiene una **arquitectura y lenguaje propio** para sus proyectos.

2.3 Dispositivo final

Este proyecto se ha desarrollado para ser jugado desde un **ordenador**, ya que se planteó que para las **mecánicas del jugador** no se podría jugar desde una plataforma como un dispositivo móvil. Se ha creado un ejecutable tanto para sistemas operativos **Windows** como para **IOS**.

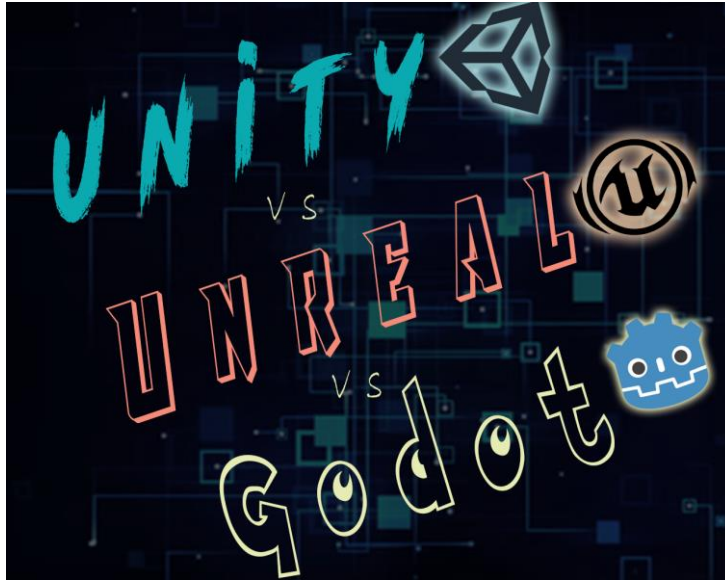


Figura 2. Unity vs Unreal vs Godot

3. Definición del juego

3.1 Breve descripción del juego

Es un juego de **acción-aventura** de un solo jugador donde se pone en la piel de un joven el cual su **padre desapareció** hace años. Se tiene que ir encontrando **pistas** y acabando con muchos monstruos mientras el pueblo sufre un constante acoso de **terremotos**. Será el protagonista el que tendrá que **salvar este pueblo** de su destrucción.

Se tendrá que ir **explorando** los distintos lugares del mapa, luchando contra los monstruos peligrosos e ir avanzando en la historia para averiguar el **secreto del pueblo** que tiembla. Cada batalla es distinta y se tiene que analizar de qué manera se actúa para no acabar muerto.

3.2 Subgénero y referencias a videojuegos existentes

Los géneros y subgéneros con los que se pueden etiquetar este videojuego son: 2D, Acción-Aventura, Rol, Exploración, Pixelart, Indie.

Algunos videojuegos de los que se podría coger referencias son: **The Legend of Zelda**, **Pokemon**, **Stardew Valley**, **Hyper Light Drifter** o **Minecraft**.

El tipo de movimiento que tiene el personaje es similar al de los antiguos juegos de “**The Legends of Zelda**” o “**Pokemon**”, ya que el personaje se puede mover en todas direcciones, pero con las limitaciones del **2D**.

El **inventario** de objetos del que dispone el jugador, igual que el funcionamiento de los cofres, es parecido a juegos de referencia como “**Stardew Valley**” o “**Minecraft**”.

Las mecánicas de lucha son parecidas al “**Hyper Light Drifter**”, ya que se puede hacer un **movimiento rápido** para esquivar ataques, se puede usar la espada para acabar con los enemigos y también se puede usar un **arco** para atacarlos a distancia.

3.3 Tipo de interacción juego-jugador

El juego pone a prueba las **habilidades del jugador** que tiene que luchar contra enemigos difíciles de abatir, pero se puede **bajar la dificultad** del juego para crear una experiencia más enfocada a la exploración y a la historia. Las mecánicas básicas del juego son **correr**, **esquivar**, **atacar** y **usar** un objeto como una poción.

De recolección: Monedas, Madera, Lingote de Hierro, Lingote de oro, gema azul y rubí.

Cuando el jugador es **abatido**, se despierta en el pueblo y todos los objetos que llevaba encima se quedan en el lugar de la **última muerte**. Estos objetos se quedan en el suelo y por lo tanto se **podrán recuperar** si se llega otra vez al lugar de la muerte.

El país está ubicado en una llanura con **bosques y montañas**, hay bosques que a veces que **impiden el paso** a ciertas áreas.



7

3.5 Historia

Cuando empieza el juego, se le dice al jugador como hablar con su madre y también que ella siempre está triste por la **desaparición de su marido**. Así ya entiendes que tu padre está desaparecido y que la historia tendrá algo que ver con él.

Esta es una ciudad muy peculiar donde muy a menudo **ocurren terremotos** y nadie sabe exactamente el motivo. Si se habla con los habitantes del pueblo, dirán que hace bastantes años que ocurren y que seguramente sea por la situación geográfica de la zona.

Justo antes de salir del pueblo se avisa al jugador que mejor coja la espada de su cofre que allí fuera hay **monstruos peligrosos**.

Cuando se sale del pueblo y se adentra en el bosque, se encuentra una nota al suelo que dice que en el **ESTE** se encontrará con más **ayuda**. Si se investiga más el bosque se verá que la salida por el **OESTE** está **bloqueada**.

Una vez se llega a la playa, si se investiga un poco el entorno, se tiene que encontrar una **nota del padre** del protagonista diciendo que ha descubierto cosas peligrosas y oscuras del pueblo y le dice que encuentre una **cueva secreta** al OESTE. Le da de regalo un **arco y flechas**.

Una vez se lea esta nota, se podrá acceder al nivel del bosque oscuro yendo más al **OESTE** del bosque. Allí habrá monstruos peligrosos y se **encontrará una cueva escondida** que solo se podrá acceder si se acaba con todos los monstruos del nivel.

Cuando se acaba con todos los monstruos y se accede a la cueva, se encuentra al **jefe de la ciudad** con tu padre **secuestrado**. Allí se tiene una conversación con ellos dos y se dice que hace años tu padre se enteró de que **los terremotos los causaban las excavaciones** de esta cueva y que lo habían **amenazado** de matar a su familia si se volvía a acercarse a alguien del pueblo.

Luego el jefe dice que los terremotos están destruyendo los cimientos del pueblo y que en unas semanas **el pueblo sería destruido**. El padre del protagonista se encara con él y el jefe se convierte en un **monstruo** que va hacia el jugador.

Allí tiene lugar la batalla final donde sale como **vencedor el protagonista** que vuelve al pueblo como un héroe y puede volver a ver a su padre y a su madre y a poder hablar con ellos.

3.6 Definición de personajes

Arant, protagonista: El personaje principal con el que se juega al juego. Es un adolescente de 16 años muy curioso y con ganas de explorar todos los rincones del país. Es fuerte y le gusta la acción. Añora mucho a su desaparecido padre, ya que él le enseñó a cazar y tiene muy buenos recuerdos con él.

Dana, madre del protagonista: La madre es una mujer muy amable que apoya a su hijo y que le da consejos de todo tipo. Está triste ya que muy a menudo piensa en su marido y en si seguirá vivo.

Rofast, padre: El padre es un hombre de mediana edad que le apasiona la caza. Siempre mira por el bien de su familia y del pueblo e intenta protegerlos a todos.

Barden, herrero: Es un personaje corpulento que se encuentra en su casa del pueblo. Te permite comprar objetos relacionados con la caza y vender todo tipo de objetos.

Mitha, bruja: Es una mujer con apariencia de bruja a la que se le puede comprar pociones.

Shax, jefe del pueblo: Es el jefe del pueblo que controla a todos los habitantes y les dice lo que pueden o no hacer. Es muy ambicioso y manipulador, no le importa sacrificar todo un pueblo a cambio de dinero y poder. Nunca ha confiado en nadie.

Estos personajes pueden tener **conversaciones** con el personaje principal. Estas conversaciones serán **frases** que dirán los personajes si presionas el botón de acción encima de ellos, te **darán consejos** para ayudarte en lo que puedas del juego.

Por ejemplo, si presionas encima de la madre del protagonista, esta te dará un consejo sobre como **explorar un área nueva** o como encontrar **notas secretas**. En cambio, con el herrero y la hechicera, cuando presiones encima de ellos, te darán un consejo sobre que **armas** son mejores que otras o sobre **pociones** y luego se abrirá un pequeño menú para poder **comercializar** con ellos.



Figura 4. Personajes del juego

3.7 Tipos de enemigos

Hay **cinco** tipos de enemigos comunes y el **jefe final**. En el nivel del bosque solo se encuentran los enemigos de nivel 1 que son los “**Murciélagos**” y los “**Goblins**”.

En el nivel de la playa se encuentran los enemigos de nivel 2, los “**Orcos**” y las “**Ratas**”. En el nivel del bosque oscuro se encuentran las Ratas y el enemigo de nivel 3, las “**Arañas**”.

Para acabar te encuentras con el “**Troll**”, que es en lo que se transforma el jefe y es el más peligroso de todos.

3.8 Objetivos planteados al jugador

Al empezar el juego, al jugador no se le dicen los objetivos finales del juego, ya que **revelarían parte del final** la historia.

Primero se le propone como **objetivos primarios**: comprar un arco, comprar pociones y aprender a cazar. Una vez aprende las mecánicas del juego se le propone: encontrar el bosque, encontrar las notas, seguir lo que dicen las notas explorando los distintos niveles.

En el menú hay un apartado de “**notas**” donde aparecen todas las notas encontradas y se pueden **volver a leer** si el jugador no ha entendido bien lo que dice o **no lo ha podido leer bien**.

La nota que se encuentra en el desierto dice que el objetivo es **mucho mayor**. Mediante las notas, va descubriendo la historia y al final se da cuenta de que las misiones que se le habían encomendado eran para conseguir **acabar con el jefe** del pueblo para que deje de explotar la mina y **salvar al pueblo** de los terremotos.

3.9 Concept Art

Está hecho con la técnica de dibujo “*pixelart*” para darle un toque **retro** y parecerse más a los juegos que se han tenido como **referencia**. Es un estilo artístico en 2D parecido al del videojuego original de “**The Legend of Zelda**” o al “**Pokemon**” con **desplazamiento de la cámara** en todas direcciones y manteniendo el jugador en **medio de la pantalla**.

Se utilizarán distintos **packs de sprites** para el diseño artístico del videojuego, siempre manteniendo la misma **estética de edad media**. Teniendo en cuenta que el pueblo está en medio de un **bosque** y hay caminos que llevan a las distintas partes del país.



Figura 5. Casa del Herrero

4. Diseño técnico

4.1 Entorno y requerimientos

Durante la planificación de todos los aspectos de jugabilidad del juego, se decidió que **debido a las mecánicas** de juego que se querían implementar, era necesario desarrollar el videojuego **enfocado** a un **entorno para ordenadores**. Teniendo en cuenta el **movimiento del personaje**, la **selección de objetos** en la barra del jugador o el **sistema de combate** creado, era difícil implementar todas esas funcionalidades para un **dispositivo móvil**. Aunque **quitando o cambiando** algunos aspectos o mecánicas del juego, se **podría haber realizado** el proyecto enfocándolo para un entorno **móvil**.

Por eso, uno de los **objetivos generales** obligatorios fue la creación de **archivos ejecutables** tanto para **Windows** como para **IOS** para facilitar la **descarga y el juego** para el jugador.

Debido a la **simplicidad de los gráficos** y a la **optimización** de las escenas y partes del juego, los **requerimientos** para poder jugar al videojuego **no son muy altas**. Con un ordenador de mesa o portátil de **gama media** se asegura su **perfecto funcionamiento**, y con uno de **gama baja** puede que no funcione **correctamente**, pero también es muy probable que **funcione** sin problemas.

4.2 Herramientas empleadas y recursos

Se han utilizado algunas herramientas para **ayudar al desarrollo** del videojuego, sobre todo han sido herramientas **ofrecidas** por el mismo motor de **desarrollo Unity**.

- Se ha usado la herramienta “**Text mesh Pro**” para la visualización y configuración de los **textos**.
- La herramienta “**Universal Render Pipeline**” para el post **procesamiento** de los **gráficos** durante la renderización del juego.
- El “**Input System**” para la gestión de las **entradas del teclado** del jugador.
- El “**Microsoft Visual Studio**” para la **organización del código** del proyecto.
- El “**Github**” como **repositorio** para **guardar y gestionar** el proyecto y sus versiones.

Se hizo un **estudio** para ver que recursos podrían ser usados para la **creación** de este proyecto y se seleccionó algunos para **ayudar** en su desarrollo. Siempre se tenía en cuenta que estos recursos tuvieran una **licencia de comercialización libre** para que se pudieran usar en este proyecto por si se colgaba en alguna **plataforma de distribución**.

Se buscaron principalmente **dos tipos** de recursos, unos paquetes de “**sprites**” o conjunto de imágenes para los **gráficos y ambientación** del juego y otros de **música y efectos** para la **inmersión** del jugador.

- **Mini World Sprites:** Usado para **recursos gráficos** e iconos de los niveles en el apartado de mapas.
<https://merchant-shade.itch.io/16x16-mini-world-sprites>
- **Pipoya free RPG Tileset 16x16:** Es un conjunto de imágenes que se utilizan para **decorar** los distintos **escenarios** del videojuego. Se ha usado para el suelo, casas, decoración, bosques y playa.
<https://pipoya.itch.io/free-rpg-tileset-16x16>
- **Kyrise's 16x16 icon pack:** Se ha usado para la mayoría de **objetos** del juego, tanto recursos, como espadas, arcos, flechas y pociones.
<https://kyrise.itch.io/kyrises-free-16x16-rpg-icon-pack>
- **The humbe item pack:** De este paquete solo se ha usado el sprite de la **nota**.
<https://thewisehedgehog.itch.io/thip>
- **Character Sprite Maker:** Este recurso ha sido muy útil para crear **distintos personajes** que estuvieran hechos con el **mismo estilo**, pero con un toque personal para identificarlos. Algunos sprites de personajes han sido **retocados manualmente**.
<https://emily2.itch.io/character-sprite-maker-16x16>
- **WonderDot Monsters pack:** Es un paquete de **recursos de pago** que proporciona varios conjuntos de sprites para dar vida a los **monstruos del videojuego**.
<https://pita.itch.io/rpg-monster-pack>
- **Pixel 16x16 heart:** Es un conjunto de **dibujos** de un **corazón** para señalar la vida restante del protagonista.
<https://pixelsapphire.itch.io/pixel-16x16-heart>
- **Joystix Monospace Font:** Es la **fuentes** utilizada para todos los **textos** del videojuego.
<https://www.dafont.com/joystix.font>

- **Medieval Song Village Consort:** Es la canción usada para el **menú** principal del juego.
https://www.youtube.com/watch?v=eZ_r1H9vHkl&list=PLnuhlOfdJ2WKvlu8eaF3uel9p3VjuK10M&ab_channel=AlwaysMusicAlwaysMusic
- **The Road Home:** Esta canción se usa para cuando el jugador está realizando el **tutorial**.
<https://www.youtube.com/watch?v=aOWAXWDfEPs>
- **Bonfire by Alexander Nakarada:** Es una canción más bien tranquila que se usa en el **primer nivel** del videojuego.
https://www.youtube.com/watch?v=WQ4lp_iAtJQ&list=PL5dOPPlvx4K8IIzaBbh8F89iWO6XObsil&index=9&ab_channel=RoyaltyFreeMusic-AlexanderNakarada
- **We're Back to Fight:** En el **segundo nivel** se usa esta canción para mejorar la inmersión del jugador.
https://www.youtube.com/watch?v=c8-JgPzv7lIQ&ab_channel=AlexanderHoff
- **The vikings by Alexander Nakarada:** Es una canción con **ritmo** definido para dar la sensación de que algo **extraño** se acerca. Se utiliza en el **tercer nivel**.
<https://www.youtube.com/watch?v=C-3e-g9s4G8>
- **Viking War Music:** Esta canción **más oscura** y fuerte es usada para el **cuarto nivel** donde hay enemigos más **difíciles**.
https://www.youtube.com/watch?v=Te6PLicFvsl&ab_channel=NoCopyrightMusic
- **Dragon Castle by Makai Symphony:** Es una canción extremadamente **épica** y **bélica** para la **batalla final** contra el jefe del pueblo.
https://www.youtube.com/watch?v=9gBTKiVqprE&ab_channel=BreakingCopyright%E2%80%9494MusicForVideos%26Streams
- **Freesound:** Para los **efectos sonoros** del videojuego como apretar un botón, seleccionar un objeto, atacar, disparar una flecha o recoger un objeto del suelo, se usó una **librería digital** de sonidos para encontrar los que mejor resultado pueden dar.
<https://freesound.org/>

4.3 Arquitectura del juego

Los archivos del proyecto están **ordenados** por **funcionalidades** y **tipos** dentro de distintas carpetas. Están ordenadas con **números** al principio de su nombre para que siempre tengan el mismo orden en todos los proyectos que se realicen y así **encontrar más fácilmente** los archivos que se están buscando.

El orden de las carpetas diferenciadas por categoría es el siguiente:

1. **Escenas** del juego.
2. archivos de código.
3. **Imágenes** usadas.
4. **Sprites** de los personajes, enemigos, interfaz y decoración.
5. **Animaciones** creadas con los sprites anteriores de los personajes, enemigos y del cofre.
6. **Objetos prefabricados** ya creados para ser utilizados más fácilmente.
7. Configuración de las distintas **entradas** del teclado.
8. Las **músicas y sonidos** que se utilizan.
9. **Fuentes** de texto usadas en el proyecto.
10. Los archivos de **recursos** que se usan para el **funcionamiento del juego** para leer o guardar variables.

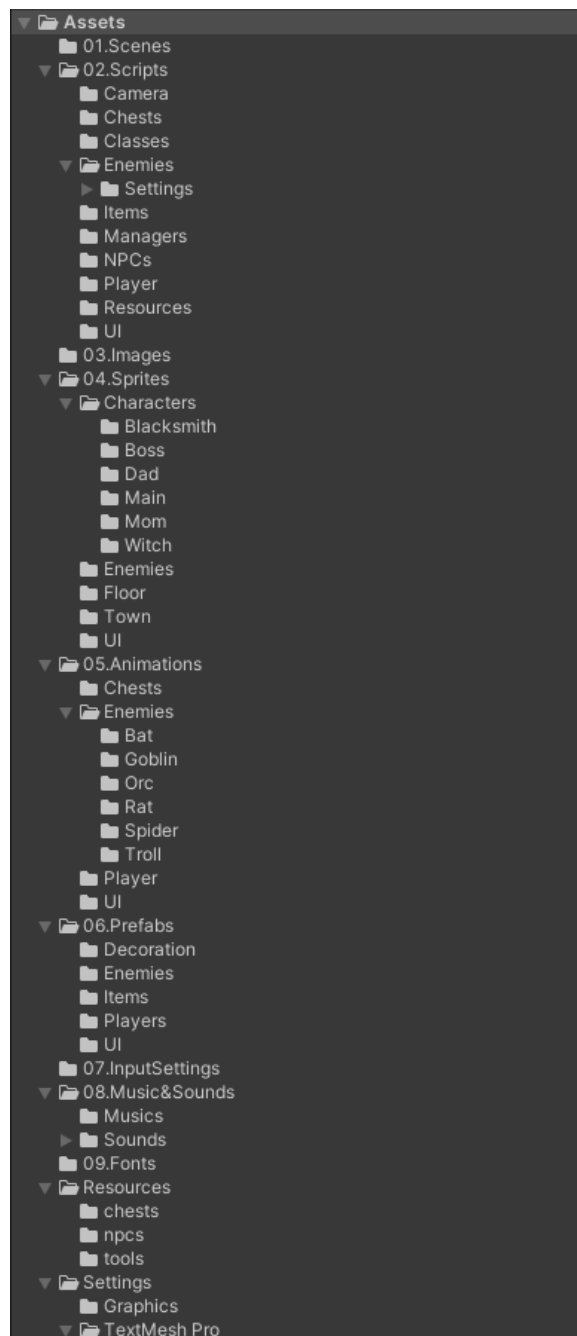


Figura 6. Carpetas dentro de "Assets"

11. **Archivos de configuración** de las herramientas.

Por encima de esta carpeta con los **archivos creados** para el desarrollo del videojuego, se encuentran las otras carpetas con la **información** para que el **proyecto** funcione con el motor de desarrollo **Unity**.

Estas carpetas constan de la carpeta anterior “**Assets**” como una de las más importantes, ya que contiene el proyecto creado por el **desarrollador**, la carpeta “**Library**” donde se encuentra la **caché** de las librerías usadas, la carpeta “**Logs**” para saber los **módulos usados** y sus versiones, la carpeta “**obj**” para **depurar** el proyecto e identificar y corregir los errores, la carpeta “**Packages**” que contiene los **plugins**, la carpeta “**ProjectSettings**” para guardar la **configuración** que se usa en el proyecto, la carpeta “**Temp**” que guarda archivos **temporales** para agilizar la lectura del programa y la carpeta “**UserSettings**” que guarda la **configuración** que usa el **usuario** al usar el programa.

Se ha de tener en cuenta que algunas de estas carpetas **no son necesarias** para abrir correctamente el proyecto, pero **agilizar** su proceso guardando cierta información. Las carpetas que se pueden **ignorar** al colgar el proyecto en un repositorio son la “**Library**”, “**obj**” y “**Temp**”.

Una carpeta muy importante que es la que permite **guardar la información del jugador** cuando abandona la partida y que pueda retomar el juego en el momento que lo dejó, es la carpeta de “**Resources**” que se encuentra dentro de la carpeta “**Assets**”.

En esta carpeta se encuentran **distintos archivos** para guardar información de la **partida**. Como en el archivo “**tools_bar**” se encuentran los **objetos** que lleva el jugador en la **barra de objetos** al guardar o en el archivo “**player_settings**” se encuentra el **nivel**, la **posición**, la **vida**, la **vida máxima** permitida, la **dificultad** del juego, el **volumen** de la **música** y el **volumen** de los **efectos sonoros**.

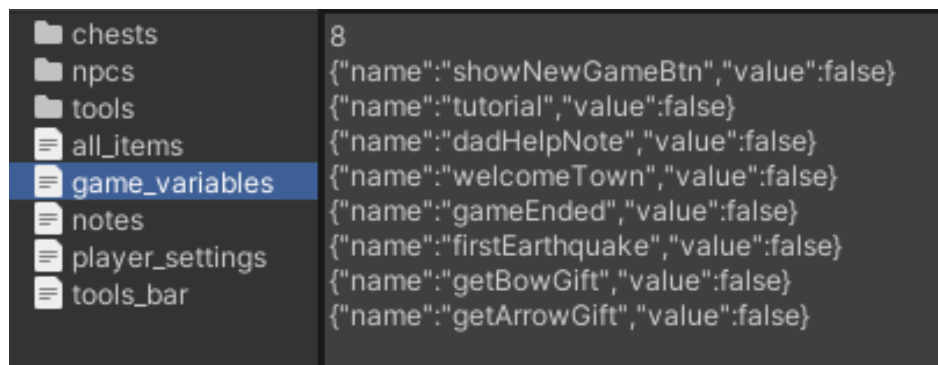


Figura 7. Archivos de la carpeta de “Resources”

4.4 Funcionamiento de las Inteligencias Artificiales

Para este videojuego se ha desarrollado una pequeña **Inteligencia Artificial** para el comportamiento de los **enemigos** en distintos escenarios posibles con los que se encuentren. Estas **secuencias de órdenes** han sido programadas de forma totalmente **configurable** con un archivo de configuración **externo**. De esta manera, es fácilmente **escalable** y mucho más fácil añadir **enemigos nuevos** con distintas características creando simplemente un **sistema de animaciones** y un **archivo de configuración** específico.

El **algoritmo** se basa en que los enemigos tienen **tres estados**, uno de **caminar**, uno de **perseguir** corriendo al enemigo y otro de **atacar**.

Predeterminadamente, el enemigo empieza con el **estado de caminar** que significa que **busca puntos** aleatorios cercanos al punto inicial en el que ha aparecido y **dentro de un radio** y va hacia ese **punto**. Si llega al punto, **busca otro** y sigue caminando, aunque si pasa demasiado tiempo y aún no ha llegado al destino, también **cambia de punto**, ya que puede significar que se ha **encallado**.

Mientras está caminando, si el jugador **entra dentro** de su **radio de detección**, el enemigo **irá corriendo** hacia él, hasta que el jugador **vuelva a salir** de ese radio o hasta que **salga** del nivel.

Como el enemigo **va corriendo** hacia el jugador, él tendrá que evitar que le **toque**, ya que si lo toca pasará al **estado de atacar**. Lo que significa que el enemigo hará la **animación de su ataque** y **quitará la vida** que sea, dependiendo del enemigo, al jugador.

Para terminar, la variable **“Difficulty Level”**, determina el **nivel del enemigo** al que se enfrenta y por tanto el nivel de los **recursos que dejará** al suelo al morir. Cuanto **más alto** es nivel, **mejor** serán las **recompensas** que deje al morir.



Figura 8. Archivo de configuración del murciélago

5. Diseño de niveles

5.1 Tutorial

El primer nivel que se encuentra el jugador es el nivel de **aprendizaje** o tutorial para interiorizar las **mecánicas básicas** del personaje y no dejar al jugador en medio de un mapa sin haberle explicado nada.

Para no despistar al jugador y que encuentre las cosas que tiene que aprender fácilmente, se ha creado un **mapa horizontal** sin posibilidad de dirigirse hacia arriba o abajo, solo de avanzar hacia delante.

Allí el jugador aprende que hay **notas** en el suelo que le van proporcionando ayuda y también ve como **moverse por el mapa**, hacer **movimientos rápidos**, abrir **cofres** para coger objetos, atacar con una **espada**, atacar con un **arco** y que los monstruos dejan caer **recursos** a la hora de morir.

Cuando el jugador llega a la otra punta del mapa, se **termina el tutorial** y aparece en el pueblo donde empieza la historia del juego.



Figura 9. Mapa del tutorial

5.2 Pueblo

El primer nivel real que se encuentra el jugador es el pueblo del protagonista. Es un sitio donde se siente **protegido y a salvo**. Su casa se encuentra a la izquierda del nivel y a la derecha se encuentran las casas de los demás habitantes del pueblo, el herrero, el jefe y la bruja.

La casa del **herrero** es la que se encuentra más arriba de las tres, allí se pueden **comprar armas y vender** todo tipo de objetos. La segunda es la casa del **jefe del pueblo**, que se puede hablar con él. Y la última es la casa de la **bruja** donde se pueden **comprar pociones**.

Dentro de la casa del protagonista hay un **cofre** donde se pueden guardar varios objetos que no se deseen perder y su **madre** se encuentra al lado, para que se pueda hablar con ella cuando se desee.

La entrada al siguiente nivel se encuentra siguiendo el camino asfaltado que va hacia el **norte**.



Figura 10. Nivel 1 - Pueblo principal

5.3 Bosque

El segundo nivel es un nivel donde el jugador tendrá que **explorar** donde está el secreto para acceder al nivel del Bosque oscuro donde se encuentra la **cueva secreta**.

Justo entrar al nivel, ya se informa al jugador que tiene que ir hacia el **Este**, donde se encuentra la playa. Si el jugador investiga hacia donde puede ir, verá que el único destino es la playa, ya que la entrada del **Oeste está bloqueada** por un bosque.

Este nivel es en el que los enemigos son más **fáciles de abatir**, ya que es el primer nivel con enemigos que se encuentra el jugador.

Una vez que encuentre la nota del padre del protagonista en el nivel de la playa, **se abrirá** una parte del bosque que hay al Oeste y se podrá acceder al siguiente nivel.

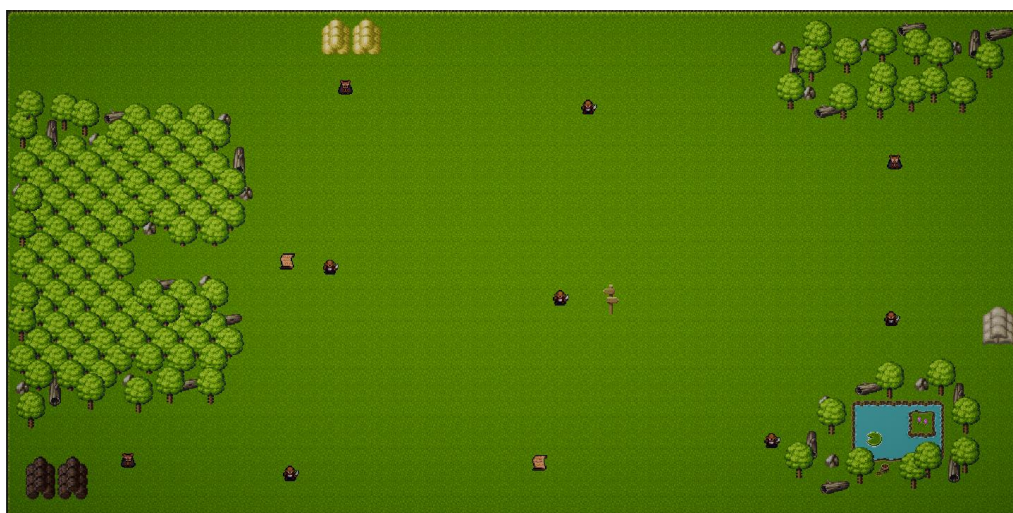


Figura 11. Nivel 2 - Bosque

5.4 Playa

En este nivel, los enemigos ya empiezan a ser **más fuertes** y se tiene que ir con cuidado, ya que al principio del juego el jugador tiene **pocos corazones** y unos pocos golpes pueden acabar con su vida.

El jugador tendrá que buscar la **nota** que ha dejado el padre del protagonista donde le explica que el pueblo está en peligro y que se dirija al **Oeste** y busque una **cueva escondida**.

El bosque que impide el paso en el anterior nivel solo se **abrirá cuando se lea esta carta**. Al lado de la carta hay **dos objetos** tirados al suelo que son regalos, solo aparecerán allí una vez por partida, cuando se recojan **no volverán a aparecer**.

Una vez leída la nota, el jugador tendrá que volver a dirigirse al nivel del bosque.

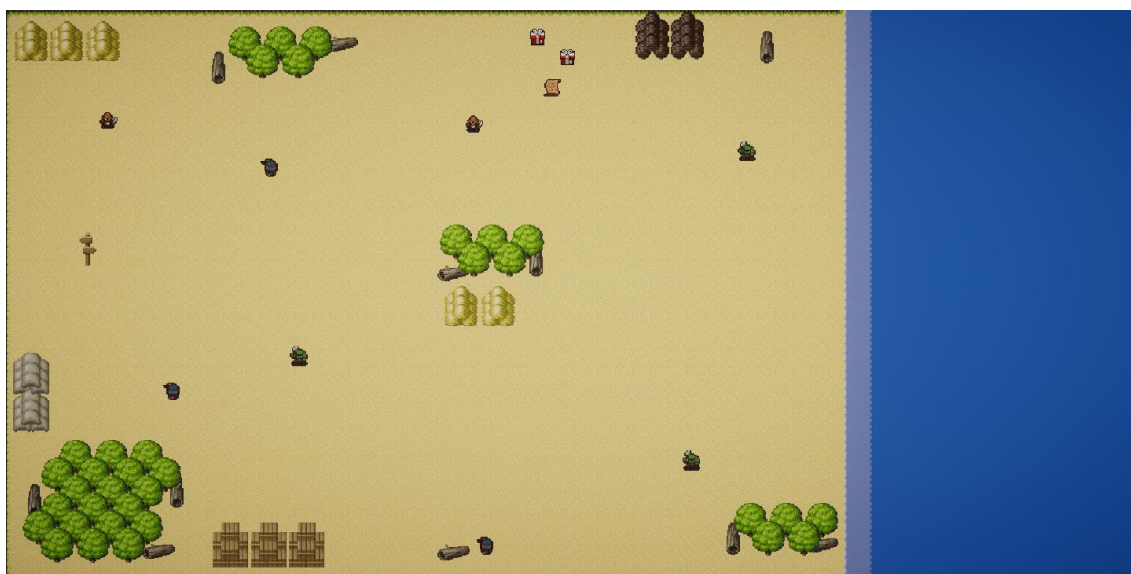


Figura 12. Nivel 3 - Playa

5.5 Bosque Oscuro

El nivel más repleto de **monstruos fuertes** que perseguirán al protagonista hasta acabar con él. Es un nivel muy difícil, ya que se tiene que buscar donde se encuentra la **cueva secreta** esquivando o acabando con los monstruos que se encuentre por en medio.

Una vez se intenta entrar a la cueva, si no has acabado con todos los monstruos del nivel, aparece una nota que indica que para acceder al nivel final se **tienen que matar todos los monstruos** que hay en el bosque oscuro. Se tiene que tener en cuenta que, si se vuelve al nivel anterior y se vuelve a entrar al bosque oscuro, los **monstruos volverán a reaparecer** y se tendrán que volver a matar.



Figura 13. Nivel 4 - Bosque Oscuro

5.6 Cueva Secreta

El último nivel, el nivel donde se **descubre la verdad** del pueblo y lo que ha estado haciendo el jefe desde hace mucho tiempo. Allí te encuentras con **tu padre** que está preso en una esquina del escenario y **al jefe** del pueblo esperándote.

Se tiene una **conversación** con estos tres personajes donde se explica lo sucedido durante años en el pueblo y la verdad sobre la **desaparición** de su padre.

Cuando se termina la conversación, el jefe se convierte en un **monstruo** que estaba esperando el momento de **revelarse**. Una vez es **derrotado**, se vuelve al pueblo con su padre y su **madre** se emociona.



Figura 14. Nivel 5 - Cueva secreta

6. Manual de usuario

6.1 Instalación

Para que se pueda **descargar** tanto el proyecto de **Unity** como los **ejecutables** de **Windows** e **IOS** se tiene que tener acceso al siguiente **repositorio**:

<https://github.com/Miquelbellet/TFM>

Una vez se haya accedido al repositorio se verán un conjunto de carpetas que son donde está ubicado el proyecto para ser **ejecutado con el programa** Unity y la versión 2020.4f1.

También hay dos **archivos extraordinarios** que son los **ejecutables** para poder jugar al juego cómodamente como una versión final del proyecto sin la necesidad de tener el programa Unity.

El archivo “**ShakyTownInstaller.exe**” es el ejecutable para instalar el juego en sistemas operativos **Windows**.

El archivo comprimido “**iosGame.exe**”, es el archivo que se tiene que **descargar** si se quiere jugar al juego en un sistema operativo **IOS**.

Simplemente con descargarse el archivo indicado dependiendo del sistema operativo, ya se podrá **jugar sin problemas** al juego.

Puede ser que el protector antivirus del navegador o del mismo sistema operativo advierten de que el archivo **no es seguro**. Esto es porque se está descargando un **ejecutable** pero no tiene **ningún programa malicioso** ni nada por el estilo dentro del archivo, simplemente contiene el juego.

En el caso de que se descargue el instalador para **Windows**, al abrirlo, se abrirá **un instalador** y se seguirán las indicaciones para que se instale correctamente el videojuego. Es preferible seleccionar el apartado de “**Crear acceso directo en el escritorio**” para poder acceder más cómodamente al juego sin tener que abrir la carpeta contenedora cada vez que queramos jugar.

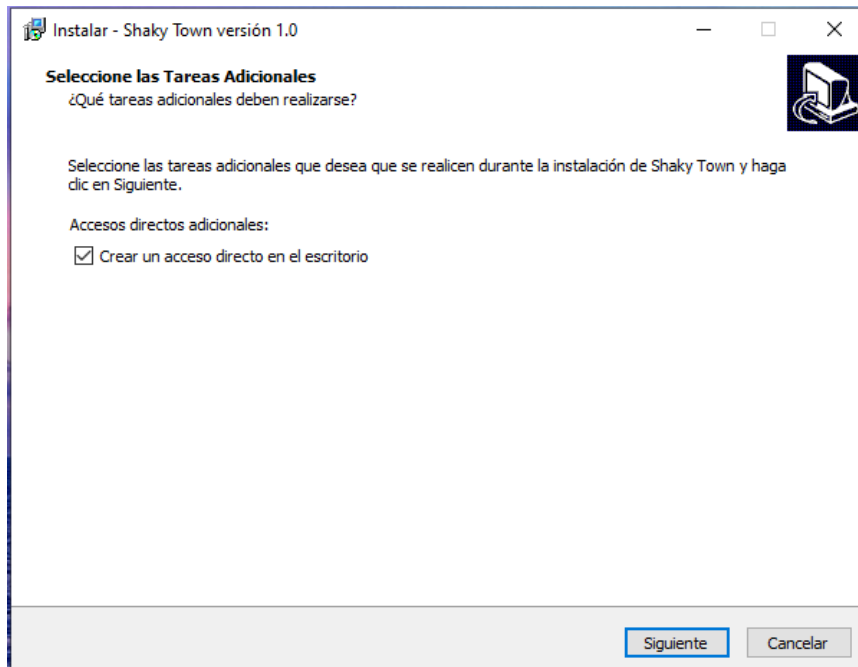


Figura 15. Instalador de Windows

6.2 Controles

Para el **movimiento** del personaje se usan los botones ["W", "A", "S", "D"] y el botón ["Shift Izquierdo"] para hacer un **movimiento rápido** hacia la dirección en la que se está **mirando**.

Usa la ["rueda del ratón"] para **cambiar de objeto** en la barra de objetos del jugador. Si se tiene seleccionada una espada o un arco y hay flechas en la barra, pulsando el **botón izquierdo** se puede atacar a los enemigos.

Con la ["E"] se puede activar la **acción del objeto** al que se está cerca, sea para **habla** con un habitante del pueblo o para abrir y cerrar un **cofre**.

Con el botón ["G"] se puede **lanzar** el objeto que se tenga seleccionado en la barra de herramientas **al suelo**. Al lanzarlo, este **desaparece** de la barra de herramientas del jugador y si se pasa por encima se puede **recoger del suelo** y se vuelve a poner en la **barra**.

7. Conclusiones

7.1 Conclusiones del trabajo

Una vez acabado este trabajo final, se ha llegado a **diferentes conclusiones**, unas más importantes que otras, pero útiles en el campo del **desarrollo** de videojuegos y de la **programación**.

Cabe decir que este trabajo no es un trabajo de investigación sino de **desarrollo de un proyecto** y que por lo tanto las conclusiones no han sido aspectos importantes para este campo sino más bien en aspectos importantes para el **crecimiento personal**.

Una de las cosas que se ha aprendido gracias a la realización de este videojuego, ha sido que una buena **planificación** puede ser la diferencia entre entregar un **producto acabado** o una demo de tus objetivos.

Es muy importante pensar claramente en las **tareas** que se tendrán que realizar a lo largo del desarrollo, en que puntos se tardará más tiempo y dejar más margen, saber qué aspectos se harán desde cero y cuáles se buscarán **paquetes de recursos**, cuáles han de ser las características básicas que ha de tener el juego y cuáles **se pueden quitar** si no hay suficiente tiempo o calcular aproximadamente el tiempo que se tardará en realizar un **conjunto de tareas**.

Otra conclusión a la que se ha llegado es que cuando se está realizando un proyecto relativamente **grande** y más de forma **autónoma**, no se tiene que tener miedo a usar **paquetes de recursos** de terceros para que sirvan de **ayuda** al desarrollador.

Se tiende a querer hacer todos los aspectos del juego **desde cero** y de forma **manual** y a veces se invierte demasiado tiempo en alguna tarea que se podría hacer más **fácil y rápidamente** con la **ayuda** de recursos.

El último aspecto que se ha visto que ha sido muy importante a la hora de realizar el proyecto, es **programar y desarrollar** las funcionalidades del juego de una forma **generalizada y configurable** para que sea un proyecto totalmente **escalable** y se puedan implementar nuevas funcionalidades fácilmente en un **futuro**.

Para finalizar, decir que se ha aprendido mucho desarrollando este videojuego, tanto en el ámbito de **gestión del proyecto** como de **programación**, diseño **gráfico** o diseño de **niveles** y que con cada proyecto que se hace se siguen aprendiendo cosas nuevas y sacando mejores conclusiones.

7.2 Reflexión sobre los objetivos planteados

Al principio del proyecto, cuando se estaba realizando la **planificación**, se plantearon unos **objetivos generales** que había de tener el proyecto en su fase **final** de entrega. Fueron unos objetivos **simples y claros** que podía abarcar muchos aspectos distintos del videojuego a realizar.

De estos aspectos generales se han **cumplido la mayoría** de ellos, como que el videojuego tuviera varios **niveles** con los que jugar, un **sistema de objetos** que el jugador pudiera usar o guardar en **cofres**, que se usaran recursos para **comprar** distintas armas en una **tienda**, que el juego tuviera distintos **niveles de dificultad** o que se pudiera jugar al juego simplemente con **un ejecutable**.

Un objetivo importante que no se ha llegado a realizar, ha sido **colgar** el videojuego en una **plataforma distribuidora** como “**steam**” o “**itch.io**”. No se ha podido realizar por **falta de tiempo**, ya que se dejó este apartado para el final y no se puso como prioritario. También cabe decir que el juego, aún siendo totalmente funcional y jugable, no tiene una **bonita presentación** ni tiene un **marketing** trabajado. Por lo que, llegado un punto, se eliminó esta opción de los planteamientos.

Como es normal, en el proceso de desarrollo de un videojuego, se van **modificando aspectos de jugabilidad**, de la **historia** o de **funcionalidades** que podrá realizar el jugador. De estos aspectos se creó un listado con las más **importantes** y las que no eran **necesarias**, dentro de **cada aspecto** general del juego.

Por ejemplo, se quitó por falta de tiempo un objeto que era una **bomba** que permitía encontrar **habitaciones secretas** en los distintos niveles. Pero al final se implementaron las **pociones**, ya que eran más fáciles y rápidas de programar.

7.3 Seguimiento de la planificación y metodología

La planificación que se **organizó** durante la preparación del proyecto, fue la que se **siguió** durante todo el **proceso** de desarrollo. Siempre teniendo en cuenta que cada punto de la planificación general tenía varios **puntos obligatorios** y varios **puntos optativos** por si no se llegaba a tiempo para la **finalización**.

También cabe añadir que la planificación se consiguió seguir gracias a que cuando el proyecto se **encallaba** en algún punto, se realizaban **horas extras** no planteadas para seguir con la **planificación inicial** hecha y no dejar de hacer puntos importantes al final del proyecto.

La metodología seguida ha sido la metodología **Scrum**, que se basa en crear **pequeños grupos de tareas** para realizar en un periodo **corto** de tiempo y así ir desarrollando los aspectos del juego.

Esta ha sido la metodología que se ha **seguido** durante todo el proceso de desarrollo, el único problema que se ha tenido, ha sido a la hora de crear **tareas individuales y cortas**. Se han ido creando tareas **demasiado grandes** para ser realizadas en un periodo corto de tiempo. Se ha tenido que hacer **algún cambio** en alguna de las implementaciones del juego para poder hacerlo más **intuitivo, simple** y acabar el proyecto a tiempo. Pero los cambios realizados no han sido a **nivel general** del proyecto, sino de **jugabilidad** del juego o de posibles **acciones** del jugador.

7.4 Líneas de trabajo futuro

Una de las **futuras implementaciones** que se realizarán para este proyecto, es uno de los **objetivos generales** planteados que no se ha podido realizar. **Subir el proyecto** a una web de **distribución** de videojuegos como "**itch.io**" para dar **visibilidad** al proyecto y que se pueda descargar fácilmente. Pero primero se tiene que trabajar el tema de la **estética, icono del juego y videos de presentación**.

También hay distintos aspectos de **jugabilidad** que se pueden mejorar, como el **sistema de combate** con espada, que actualmente no es intuitivo ni agradable para el jugador. O las **animaciones del personaje**, tanto las de movimiento como el movimiento rápido, que se podrían hacer **más completas** y así mejorar la reacción que tiene el jugador al **interactuar con el personaje**.

Se podrían añadir **más objetos** con los que el jugador pueda interaccionar, más objetos de **recursos** o más **enemigos**, pero si se quiere **ampliar tanto** el videojuego, primero se tiene que **ampliar la historia**. Por eso, uno de los aspectos que se mejorarían si se trabajara más en este proyecto sería hacer una **historia más detallada y larga** para permitir añadir **más niveles, enemigos y armas**.

8. Glosario

Diagrama de Gantt: Es una herramienta gráfica que te ayuda a tener una vista general de las tareas programadas y la fecha en que deben estar acabadas.

Scrum: Se trata de una metodología ágil que consiste en la entrega de tareas en periodos cortos de tiempo y las tareas se van adaptando en función del desarrollo del proyecto.

GDD (Game Design Document): Es un documento muy descriptivo que se va actualizando por el equipo de desarrollo de un videojuego. Sirve para organizar el equipo y para usarlo de guía durante el proceso.

Mecánicas de juego: Las mecánicas son cualquier acción que realice el jugador que altere el estado del juego, sea el movimiento, características, objetos que se usan, etc.

Game Engines o motores de desarrollo: Un motor de desarrollo son un conjunto de librerías y funcionalidades que se usan para el diseño, creación y representación de un videojuego.

Pixelart: Es una forma de arte digital, creada a través de un ordenador y un programa de creación. Las imágenes resultantes son píxeles individuales que se juntan el bloque para crear la imagen.

Videojuegos triple A: Son aquellos videojuegos desarrollados por las grandes compañías con costes altos tanto en el desarrollo como en el marketing.

Sprite: Es una imagen de bits presente en la pantalla, una máscara de un objeto que tiene la capacidad de colisión.

Repositorio: Es un espacio centrado en el almacenamiento y organización de información digital que pueden contener trabajos de software

Paquetes de recursos: Un grupo de carpetas que puede contener distintos tipos de archivos que ayudan al usuario con el desarrollo de un proyecto.

Tileset: Es un conjunto de imágenes que se usan para completar escenarios de un videojuego. Cada videojuego puede tener varios separados por lugares, montañas, lagos, etc.

Script: Es un archivo de texto donde se guardan partes de código con secuencias de instrucciones que son interpretadas por el compilador.

9. Bibliografía

1. Andrew Beattie (2020) - <https://www.investopedia.com/articles/investing/053115/how-video-game-industry-changing.asp> (30/05/2021)
2. Lindsay Schardon (2021) - https://gamedevacademy.org/best-game-engines/#GameMaker_Studio_2 (30/05/2021)

10. Anexos

No hay anexos complementarios.