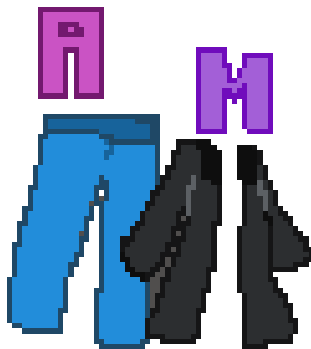


# Mario Vaquerizo perdió su Chaqueta



Autor: Enrique López Celma.

Tutor: Vicent Moncho Mas.

Profesor: Pau Magrané Figuera.

# FICHA DEL TRABAJO FINAL

|   |   |
|---|---|
| <b>Título del trabajo:</b>  | <i>Mario Vaquerizo perdió su Chaqueta</i> |
| <b>Nombre del autor:</b>  | <i>Enrique López Celma</i>                |
| <b>Nombre del colaborador/a docente :</b>   | <i>Vincent Moncho Mas</i>                 |
| <b>Nombre del PRA:</b>  | <i>Pau Magrané Figuera</i>                |
| <b>Fecha de entrega (mm/aaaa):</b>  | <i>03/2023</i>                            |
| <b>Titulación o programa:</b>   | <i>Plan de estudios</i>                   |
| <b>Área del Trabajo Final:</b>  | <i>Videojuegos</i>                        |
| <b>Idioma del trabajo:</b>  | <i>Castellano</i>                         |
| <b>Palabras clave</b>   | <i>Unity, Plataformas 2D, Aventuras</i>   |
| <b>Resumen del Trabajo (máximo 250 palabras):</b> <i>Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo</i>   |   |
| <p>La realización de este trabajo final pretende, por una parte, llevar a la práctica la mayor de los conocimientos durante el grado y, por otra, aprender más sobre el desarrollo de videojuegos y el lenguaje C# que incluye Unity.</p> <p>Mario Vaquerizo es el protagonista del juego que voy a crear. El es un personaje famoso español que ha aparecido en múltiples shows y conciertos junto con Alaska, su mujer. En este videojuego trata sobre que Alaska ha sido secuestrada y para recuperarla Mario tendrá que luchar en múltiples escenarios contra Bertin Osborne y su banda. Ellos han sido quienes han secuestrado a Alaska. Como jugadores controlaremos a Mario y tendremos que avanzar pantalla a pantalla derrotando a todos sus enemigos para poder recuperar a su amada Alaska.</p> <p>El desarrollo se llevará a cabo en dos programas. La parte gráfica para hacer los personajes y las animaciones de los mismos la haré en pixel art y luego llevaré los diseños e implementaré el código con Unity.</p> |   |
| <b>Abstract (in English, 250 words or less):</b>  |   |

The realization of this final project aims, on the one hand, to put into practice the greatest knowledge during the degree and, on the other to learn more about the development of video games and the C# language that Unity includes.

Mario Vaquerizo is the protagonist of the video game that I am going to create. He is a famous spanish character who has appeared in multiple shows and concerts along with Alaska, his wife. In this video game it deals with the fact that Alaska has been kidnapped and to recover it. . In this video game it deals with the fact that Alaska has been kidnapped and to recover it Mario will have to fight in multiple scenarios against Bertin Osborne and his gang. They have been the ones who have kidnapped alaska. As players we will control Mario and we will have to advance screen by screen defeating all his enemies in order to recover his beloved Alaska.

Development will take place in two programs. I will do the graphic part to make the characters and their animations in pixel art and then I will take designs and implement the code with Unity.

# Índice

|  |       |
|--|-------|
| <b>1. Introducción</b>   | 6     |
| <b>1.1. Introducción</b>   | 6     |
| 1.1.1. Contexto y justificación del trabajo                          | 6     |
| <b>1.2. Descripción</b>  | 6-7   |
| <b>1.3. Objetivos generales</b>                                      | 8     |
| 1.3.1. Objetivos principales   | 8     |
| 1.3.2. Objetivos secundarios   | 8     |
| <b>1.4. Metodología y proceso de trabajo</b>                         | 8-9   |
| <b>1.5. Planificación</b>  | 9     |
| <b>1.6. Presupuesto</b>  | 9     |
| <b>1.7. Estructura del resto del documento</b>                       | 9-10  |
| <b>2. Análisis de mercado</b>  | 10-14 |
| <b>2.1. Público objetivo y perfiles de usuario</b>                   | 10-11 |
| <b>2.2. Marco teórico y estado del arte</b>                          | 11-14 |
| <b>2.3. Análisis DAFO</b>  | 14-15 |
| <b>3. Propuesta</b>  | 15-17 |
| <b>3.1. Definición de objetivos/especificaciones del producto</b>    | 15-16 |
| <b>3.2. Modelo de negocio</b>  | 16    |
| <b>3.3. Estrategia de marketing</b>                                  | 17    |
| <b>4. Diseño</b>   | 17-30 |
| <b>4.1. Arquitectura general de la aplicación/sistema/servicio</b>   | 17-18 |
| <b>4.2. Arquitectura de la información y diagramas de navegación</b> | 19-21 |
| <b>4.3. Diseño gráfico e interfaces</b>                              | 22-28 |
| 4.3.1. Estilos   | 29-30 |
| 4.3.2. Usabilidad /UX  | 31-32 |
| <b>4.4. Lenguajes de programación y APIs utilizados</b>              | 32-33 |
| <b>5. Implementación</b>   | 33-37 |
| <b>5.1. Requisitos de instalación</b>                                | 33-34 |

---

|   |       |
|---|-------|
| <b>5.2. Instrucciones de instalación</b>  | 34-37 |
| <b>6. Demostración</b>                    | 37-42 |
| <b>6.1. Instrucciones de uso</b>          | 37-38 |
| <b>6.2. Prototipos</b>                    | 38-42 |
| 6.2.1. Prototipos Lo-Fi                   | 26    |
| 6.2.2. Prototipos Hi-Fi                   | 26    |
| <b>6.3. Tests</b>                         | 42-44 |
| <b>6.4. Guía de usuario</b>               | 45    |
| <b>7. Conclusiones y líneas de futuro</b> | 46    |
| <b>7.1. Conclusiones</b>                  | 46    |
| <b>7.2. Líneas de futuro</b>              | 46    |
| <b>Bibliografía</b>                       | 47    |

# 1. Introducción

## 1.1. Contexto y justificación del trabajo.

La idea del trabajo surgió cuando estaba cursando el grado superior de Animación 2D, 3D, creación de videojuegos y entornos interactivos. Hice unos diseños, pero lo dejé un poco en el aire y ahora quería retomarla mejorando esos bocetos previos y usando mis conocimientos de programación adquiridos a lo largo de toda la carrera.

Esta situación está unida a mi curiosidad por aprender a desarrollar videojuegos de plataformas, mi género favorito. Además, quería seguir aprendiendo sobre Unity, así que he aprovechado la excusa de elaborar el trabajo de fin de grado para adentrarme en este mundo.

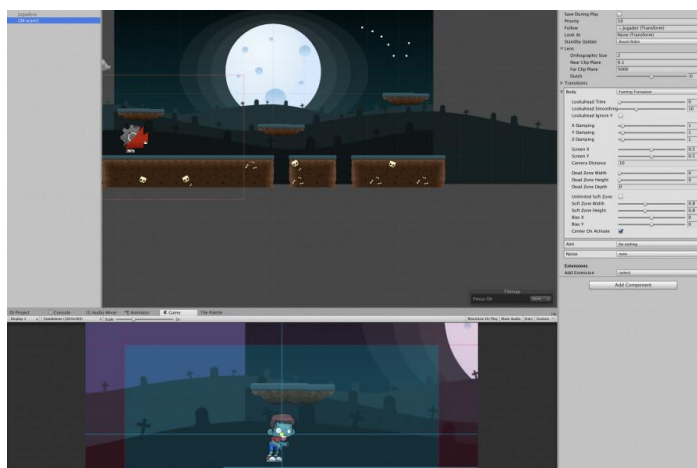


Figura 1.1: Interfaz Unity

## 1.2. Descripción.

- **Punto de partida del trabajo**

Lo primero que habrá que realizar será esquematizar los diseños de los personajes y de las animaciones de los mismos a papel. Una vez hecho esto usará el programa PixelArt para convertir este croquis en una pieza digital.

- **¿Cuál es la necesidad a cubrir?**

No sería una necesidad en concreto para solventar un problema en un colectivo, sino que es para mí, para aprender sobre cómo se debe desarrollar un videojuego desde su punto de inicio hasta su prototipo final. Es decir, conocer el proceso creativo y lógico de un juego.

- **¿Por qué es un tema relevante?**

Pienso que la creación de un videojuego siempre es interesante, porque hay muchas formas de hacerlo y cada uno tiene su metodología. Es una buena forma de trabajar múltiples áreas de la carrera y ampliando los conocimientos aprendidos en el grado.

- **¿Cómo se resuelve el problema en la actualidad?**

Las empresas profesionales elaboran un planning basado en una serie de tareas para organizarse a la hora de elaborar un proyecto. Después de esto asignan personas a cada una de las áreas a trabajar. Hay gente que se dedica a la parte visual y otros a la más interna, el código. Sin embargo, yo lo voy obrar todo yo mismo, pero sí que voy a seguir un planteamiento o una estrategia para poder organizarme y que no me pierda a la hora de desarrollar estas tareas necesarias para la elaboración del juego.

- **¿Qué resultado se quiere obtener?**

Adquirir los conocimientos necesarios para hacer una pieza audiovisual donde se presente un juego de la mejor manera posible.

- **¿Qué producto se obtendrá como resultado final de este trabajo?**

Un videojuego de plataformas 2D.

- **¿Cuáles son las características principales de estos productos?**

-Jugabilidad

-Diseño atractivo

## **1.3. Objetivos generales.**

### **1.3.1. Objetivos principales**

Objetivos de la aplicación/producto/servicio:

- Mostrar el resultado de un videojuego para ordenadores

Objetivos para el cliente/usuario:

- Divertir a los jugadores
- Mejorar las capacidades de escucha y de concentración.

Objetivos personales del autor del TF:

- Aprender a programar en Unity
- Adquirir conocimientos de desarrollo de videojuegos

### **1.3.2. Objetivos secundarios**

Objetivos adicionales que enriquecen el TF.

- Profundizar en los aspectos visuales del videojuego
- Aprender el lenguaje C#

## **1.4. Metodología y proceso de trabajo.**

Para el análisis y el diseño del videojuego se han utilizado diversas herramientas software que permiten la realización de diagramas UML (de clases, de casos de usos, de frecuencia).

En cuanto al diseño del mismo se llevará a cabo el uso habitual, basado en 3 etapas: diseño conceptual, diseño lógico y diseño físico.

Una vez dispuestos los diseños se hará un sistema kanban para la implementación, el cual nos permite utilizar tarjetas para organizar las tareas pendientes, las cuales dividiré en 3 posibles estados: pendientes, en desarrollo, finalizados.



Para facilitar el seguimiento de las tareas, utilizaré el programa Trello, con lo que puedo llevar el registro de actividades mediante tarjetas visuales.

## 1.5. Planificación

La planificación del trabajo se llevará a cabo en base al tiempo disponible para la realización del proyecto desde la fecha de inicio del semestre (01/03/2023) hasta la fecha de la última entrega (18/06/2023).

Durante dicho periodo el desarrollo se va a dividir en 4 etapas: plan de trabajo (diagrama de gantt), análisis y el diseño del producto (diagrama de gantt), implementación (diagrama de gantt) y memoria y presentación (diagrama de gantt).

## 1.6. Presupuesto

Se va realizar la siguiente distribución de costes por recurso a nivel de proyecto. Aunque algunos elementos ya estaban bajo mi disposición antes de empezar el desarrollo, no obstante, el proyecto se va a tratar como si se empezase desde 0, y por lo tanto se contabilizarán dichos elementos.

| Elemento   | Descripción   | Coste     |
|--|---|-----------|
| HP Omen, 40 L, GT21-0001NS, AMD RYZEN 5, 5600 G/16 TB/ 1TB SSD, RTX 3060 | Ordenador de sobremesa                                | 1049, 99€ |
| Unity  | Software para desarrollar videojuegos                 | 0€        |
| PixelArt   | software para animar y crear personajes con pixel art | 9,99€     |

Figura 1.2: tabla con el presupuesto de los costes.

## 1.7. Estructura del resto del documento

A continuación se realizará una breve descripción de los contenidos que abordaré en cada uno de los capítulos de los que se compone la memoria, con el objetivo de buscar búsquedas concretas de información.

- **Tecnologías utilizadas.**

En este capítulo se hará un recorrido por todas las tecnologías utilizadas tanto en las fases de diseño como en las fases de implementación del proyecto.

- **Requisitos del proyecto**

Aquí repasaré los diferentes requisitos del proyecto.

- **Diseño del proyecto**

Se podrá ver el avance del diseño tanto en el programa de diseño como en Unity.

- **Desarrollo e implementación del proyecto**

Capítulo donde se describirán los procesos de implementación.

- **Flujo de ejecución del videojuego**

Se realizará un recorrido por una ejecución finalizada del videojuego.

- **Conclusiones**

Finalmente en este capítulo se recogerá todas las conclusiones obtenidas en la elaboración del trabajo, así como sus posibles futuras posibilidades del mismo.

## 2. Análisis del mercado

### 2.1. Público objetivo

Como sabemos el mercado de videojuegos se está extendiendo mucho y cada vez crece más, se crean nuevas plataformas para poder jugarlos y es cada vez más asequible hacerse con alguna entrega de ellos. A pesar que la demanda es alta, es difícil hacerse hueco entre tantos títulos y dar a conocer nuestro juego, ya que las empresas grandes donde se producen juegos cada poco tiempo son las que se llevan casi todo el reconocimiento, pero eso no significa que no se pueda abrir un brecha en el mercado y que entre el proyecto que tenemos pensado hacer, pero es más difícil claro está. Los juegos indies se están vendiendo cada vez más y cada vez más hay más usuarios que consumen este tipo de videojuegos, pero siempre los más vendidos y los que sacan más recompensas son los videojuegos AAA. Estos han usado múltiples recursos para su desarrollo y han movido mucho el mercado del marketing anunciando su videojuego. Para un desarrollador indie o desarrolladores como nosotros es muy difícil alcanzar esos objetivos, pero podemos buscar orientar nuestro videojuego a un

público objetivo, es decir, un mini mercado donde creemos que en ese nicho se puede explotar más nuestro producto.

En cuanto a los medios de distribución, analizando el mercado y como previamente he dicho, cada vez se va ampliando más los dispositivos donde se puede jugar e incluso existen plataformas online donde los juegos se almacenan en la nube o en el propio software. Estas oportunidades van surgiendo más a menudo y esto ayuda gratamente a los desarrolladores que no tienen un estudio propio y que no tienen los mismos recursos que una gran productora. Algunos ejemplos de que si se hace un videojuego interesante se puede sacar grandes beneficios son Hollow Knight, Little Nightmares, Limbo y The Binding of Isaac.



Figura 2.1: Little Nightmares

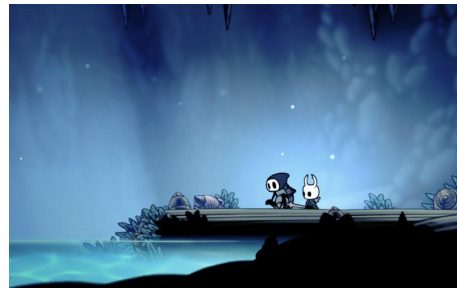


Figura 2.2: Hollow Knight

## 2.2. Marco teórico y estado del arte

Mario Vaquerizo es un videojuego de plataformas 2D. Es de género acción-aventuras, incluyendo elementos de acción como son las armas de Mario y las que tienen los enemigos, pero es una acción rápida tipo Mario Bros. También incluye elementos para interactuar de manera lógica, ya que el jugador tiene que pensar la mejor manera de pasarse la pantalla posible. Estos tipos de juego es uno de los géneros más fundamentales en la historia del videojuego convirtiéndose en el tipo de videojuego más popular en los años 80-90, el mismo que podemos encontrar en las máquinas recreativas de esos años, por ejemplo, Metal Solid. Videojuegos donde como jugador tenemos que escalar, saltar, disparar y recoger objetos mientras evitamos obstáculos que nos aparecerán a lo largo del camino. Aunque estaban en auge en los años 80-90, estos videojuegos empezaron a llegar al mercado a finales de los años 70.

El arcade **Space Panic** fue el primero en este género, no se podía mucho interactuar con los personajes, ya que no existía la opción de saltar, por lo que el juego no satisfacía las necesidades más modernas, pero influyó mucho en el género. Era un juego de subir escaleras, elemento que se puso en común en los juegos de plataformas iniciales.

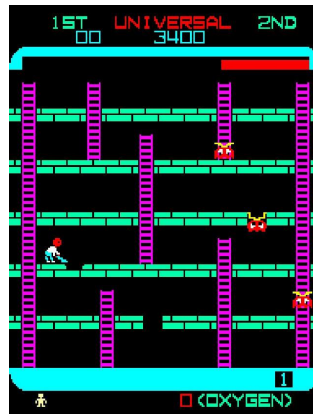


Figura 2.3: Space Panic

**Donkey Kong**, creado por Nintendo y lanzado en julio de 1981, fue el primer juego que permitió a los jugadores saltar por encima de los obstáculos y los agujeros, lo que lo convirtió en el primer verdadero juego de plataformas. El protagonista era “Jumpman” que más tarde se convertiría en **Mario**.

**Donkey Kong** fue distribuido en las consolas y ordenadores de la época. El juego ayudó a consolidar la posición de **Nintendo** como un nombre importante en la industria de los videojuegos a nivel internacional.

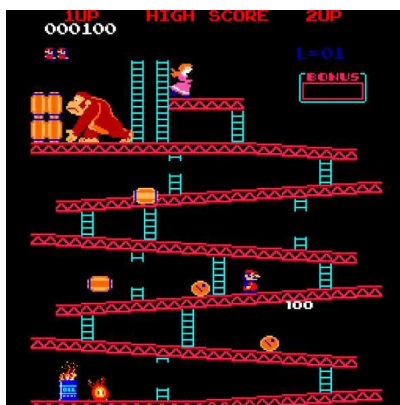


Figura 2.4: Donkey Kong.



Figura 2.5: Logo compañía de Nintendo.

Aproximadamente por esa época otros juegos como **Mario Bros**, o **Bubble Bobble** mantuvieron el estilo de plataformas estáticos verticales y además se añadió la participación de un segundo jugador.

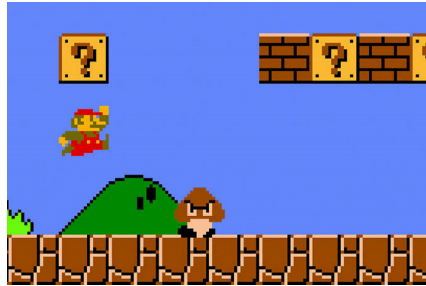


Figura 2.6: Mario Bros.

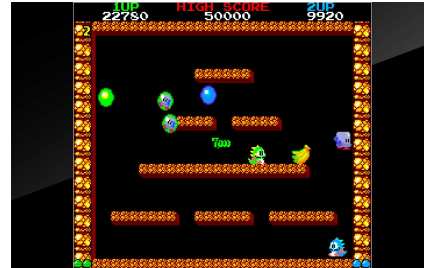


Figura 2.7: Bubble Bobble.

A partir de 1982, surgieron que tenían niveles y abarcaban varias pantallas conectadas. **Pitfall !**, lanzado para **Atari 2600**, presentaba niveles amplios y extendidos horizontalmente. Se convirtió en el juego más vendido del sistema y fue un gran avance para el género. **Manic Miner** (1983) y su secuela **Jet Set Willy** (1984) continuaron con este estilo de niveles multipantalla en ordenadores. En 1984, **Epyx** lanzó **Impossible Mission**, que amplió aún más el aspecto de exploración y sentó las bases para juegos como **Prince of Persia**.

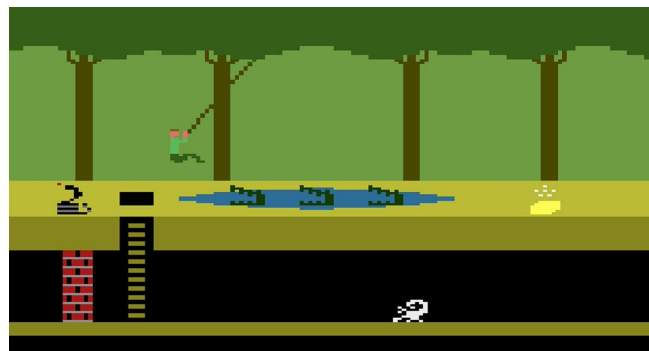


Figura 2.8: Pitfall.

Este último presentó un nivel de animación sin precedentes. Era detallada con movimientos y físicas mucho más realistas que lo visto hasta entonces. Inspiró además influyó en otros juegos cinematográficos como **Another World** o **Flashback** muy populares tanto en ordenadores como en consolas.



Figura 2.9: Prince of Persia.

La gran época que vivían los juegos de plataformas, también generó la época de las mascotas. Muchas compañías diseñaron mascotas antropomorfas buscando hacerse un hueco a la sombra de Mario y Sonic unos con más calidad que otros.

Con la eclosión de los desarrolladores independientes a través de las plataformas de venta digital, se han seguido lanzando multitud de juegos de este tipo. *Freedom Planet*, *Shovel Knight*, *Hollow Knight* o *Cuphead*, son ejemplos de plataformas muy bien recibidos por crítica y público. Se han lanzado remakes de *Wonderboy* o *Toki*. Ubisoft lanzó nuevas entregas de *Rayman* en 2D, y Nintendo ha seguido lanzando entregas de *New Super Mario Bros* y *Donkey Kong*.

### 2.3. Análisis DAFO

#### Fortalezas:

1. Ganas de aprender.
2. Contar con los recursos apropiados para realizar el proyecto adecuadamente.

#### Debilidades:

1. No conozco mucho el programa Unity.
2. Desconozco como se usa el programa pixelArt

#### Oportunidades:

1. Conseguir realizar el videojuego y subirlo a la plataforma steam

- Propuesta del videojuego como carta de presentación en una empresa.

#### Amenazas:

- No hay mucha información sobre cómo crear juegos que no supongan un coste.
- Hay juegos muy parecidos al que desarrollare, así que lo mismo resulta poco atractivo.

## 3.Propuesta

Este proyecto consiste en la creación de un videojuego indie para ordenadores y se realizará en el programa Unity. Es un juego de plataformas donde el jugador encarna el papel de Mario Vaquerizo, personaje famoso de la tele. El jugador en el transcurso de varios escenarios deberá intentar recuperar a su amada Alaska de manos de la “Organización de Participación en Eurovisión España”. Ella fue raptada a raíz de que ganó Eurovisión y como la organización pensó que no se lo merecía, la tienen encarcelada en el palacio real.

### 3.1 Definición de objetivos/especificaciones del producto.

El objetivo principal como jugadores será completar las diferentes pantallas y derrotar a todos los enemigos para llegar al jefe final “Rosa de España” que se encontrará en el palacio real, donde tiene presa a Alaska. Sin embargo, para llegar a rescatar a Alaska tendrá que derrotar a tres de los mayores ídolos de la música en España que se opondrán a que avance Mario: Raphael, Bisbal, Chikilicuatre, Las supremas de Móstoles y Rosa. Estos se distribuirán en los diferentes escenarios. Cada ídolo tendrá un escenario y para que el personaje pueda continuar tendrá que derrotar a todo el ejército de cada ídolo.

| Ídolos                   | Ciudades            |
|--------------------------|---------------------|
| Raphael                  | Linares, Jaén       |
| Rodolfo Chikilicuatre    | Igualada, Barcelona |
| Las supremas de Móstoles | Móstoles, Madrid    |
| Rosa López               | Granada, Andalucía  |

Figura 3.1: Tabla con la explicación de los diferentes personajes y escenarios del juego.

En cuanto a la funcionalidad, deben incluirse elementos como menús, etc. Por este motivo se va a realizar una planificación de objetivos que nos asegure que ningún elemento quede descuidado. Los

aspectos de jugabilidad son varios: el jugador (Mario Vaquerizo), Alaska, los enemigos y las armas. Cada uno tendrá sus componentes de sonido y animaciones y en el caso de los enemigos tendrán que disponer de una inteligencia artificial (IA).

### 3.2 Modelo de negocio

En caso de que corresponda, se incluirá un apartado de valoración económica del trabajo. Este apartado indicará los gastos asociados a la hipotética (o no) puesta en marcha del proyecto, los beneficios económicos que se esperan y la manera en que se esperan obtener, así como un análisis sobre la viabilidad económica del producto.

Gastos asociados a la programación del proyecto. Se partirá de un salario de programador junior, que es de 20€ la hora. Se estima que el proyecto tenga unas 400 horas de duración. Respecto a la parte gráfica como diseñador gráfico serán unas 100 horas aproximadamente.

| Elemento           | Descripción  | Coste  |
|--------------------|--------------|--------|
| Programador        | mano de obra | 6.000€ |
| Diseñador gráfico  | mano de obra | 2.000€ |
| Total mano de obra |              | 8.000€ |

Figura 3.2: Tabla con el presupuesto del personal desarrollador del videojuego.

Para que el proyecto sea viable económicamente necesitamos darlo a conocer. Hoy en día existen miles de juegos en el mercado con características similares por lo que es muy complicado dar a conocer el juego sin una campaña de marketing adecuada. Existen plataformas como STEAM que son utilizadas por millones de usuarios y donde podemos dar a conocer nuestro juego. Para conseguirlo se creó Steam Greenlight. Para poder realizar la instalación completa, tendrás que pagar una tarifa de \$100.00 USD por cada producto que desees vender en Steam (la "tarifa de Steam Direct"). Podrás realizar ese pago con cualquier método de pago compatible con Steam en tu región, excluyendo métodos que utilicen la Cartera de Steam.

Este pago no es reembolsable, pero se podrá recuperar en el pago efectuado después de que tu producto tenga un ajuste de ingresos brutos de \$1,000.00 USD en la tienda de Steam y en compras dentro de la app. Se retendrá el pago de los ingresos de las ventas y el reembolso de la tarifa si el pago del depósito se rechaza, devuelve o si en cualquier caso se identifica como fraudulento.



Para que nuestro juego figure dentro de Steam necesitaremos preparar una portada, 4 o más capturas, un video del juego en funcionamiento y al menos 2 párrafos que describan el juego y los requisitos de sistema. Los usuarios de Steam podrán entonces acceder a la página propia del juego y votar Si o No según lo interesados que estén en el juego. Una vez alcanzados un número suficiente de votos positivos Valve se pone en contacto con los desarrolladores para definir un calendario de entrega (si el juego no está acabado) y un precio para el producto. Este proceso tiene un coste de \$100.

### **3.3. Estrategia de marketing**

Es necesario dar a conocer el juego en todos los medios de los que dispongamos, como foros y redes sociales (Facebook, Twitter, Google +, instagram, Tik Tok...) que apunten a nuestra página de Steam. Previamente debemos evaluar los costes que suponen los posts a estas plataformas, condiciones y licencias necesarias. En el caso de que el producto final sea de calidad y estemos convencidos de su éxito podemos dedicar mayores recursos económicos en darlo a conocer a través de banners publicitarios, reviews de youtubers dedicados y otros medios.

## **4. Diseño**

### **4.1. Arquitectura general de la aplicación/sistema/servicio**

Tal y como se ha explicado en los anteriores apartados se busca un motor gráfico que ayude a ganar tiempo, con una curva de aprendizaje adecuada. Este tiene que admitir la programación en diversos sistemas. Por otro lado, tiene que ser fácil tratar recursos gráficos en 2D.

Al tener muy poca experiencia programando videojuegos el motor tiene que tener una comunidad activa, abundante en recursos y tutoriales disponibles. Por último, a nivel personal, tiene que ser un motor actual, ya que me gustaría aprender algo que fuera más allá del TFG.

Teniendo en cuenta esto, se ha escogido el motor Unity que a su vez tiene un IDE complementario para crear códigos, Visual Studio, siendo gratuito y facilitándonos mucho la tarea como programador. Además que esta herramienta es conocida y proporciona confianza al utilizarse en múltiples asignaturas anteriormente. Con este editor se realizarán los scripts que se aliaran a algunos de los componentes que hay en la escena. Estos estarán escritos en el lenguaje de programación C#.

La parte visual se desarrolla con Piskel, es una webbapp intuitiva y muy potente para crear pixel art. Cuenta con todos los elementos básicos que podemos encontrar en un editor gráfico, incluyendo

selección a mano alzada, un gotero para copiar colores y una paleta de soporte de códigos hexadecimales. La sección de configuración nos permite activar la grilla de píxeles, cambiar la transparencia de las capas y alterar la velocidad de los fotogramas por segundo. Esto significa que piskel soporta, desde animaciones básicas hasta animaciones complejas.

Aunque la mayor parte de los elementos visuales se desarrollaran en la webapp Piskel, también se realizará algunos diseños con Pixel Edit, aplicación de dibujo especialmente diseñada para trabajar con tiles. Es decir, fragmentos cuadrículados que encajan dentro de una rejilla predeterminada.

Para llevar el control sobre el código que se está empleando y sobre las modificaciones, se usará el repositorio GitHub, ya que permite alojar el proyecto utilizando un control de versiones basado en la nube que ayuda a desarrollar, a almacenar y administrar su código.

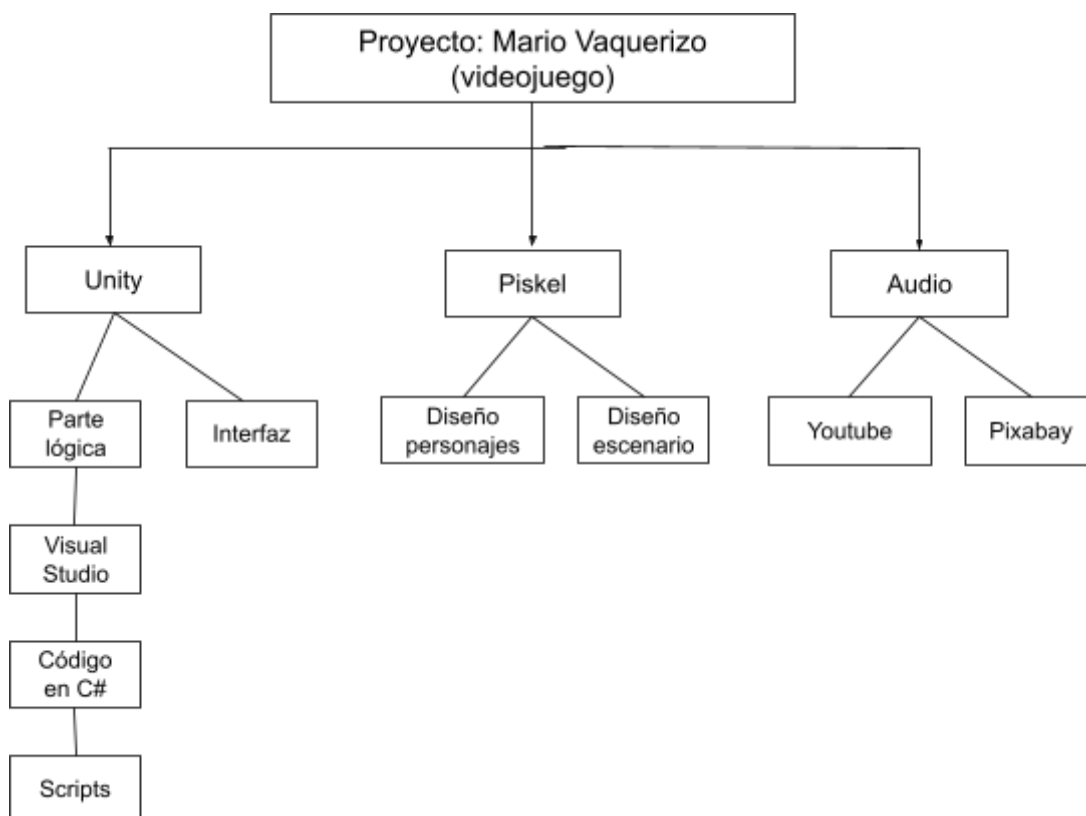


Figura 4.1 Diagrama general del videojuego

## 4.2. Arquitectura de la información y diagramas de navegación

La forma como se ha organizado el juego es a través de una división de escenas. A partir de ahí, cada escena tiene sus propios objetos de juego (GameObjects) los cuales, a su vez tienen scripts añadidos que permiten añadir lógica a dichos objetos.

Mario Vaquerizo cuenta con 4 escenas: escena de “*Raphael*”, escena de “*Las Veneno*”, escena de “*Chikilicuatre*” y escena final “*Rosa de España*”. Cada una de estas escenas tiene una serie de elementos, componentes y assets asociados a los gameObjects implementados dentro de ellas.

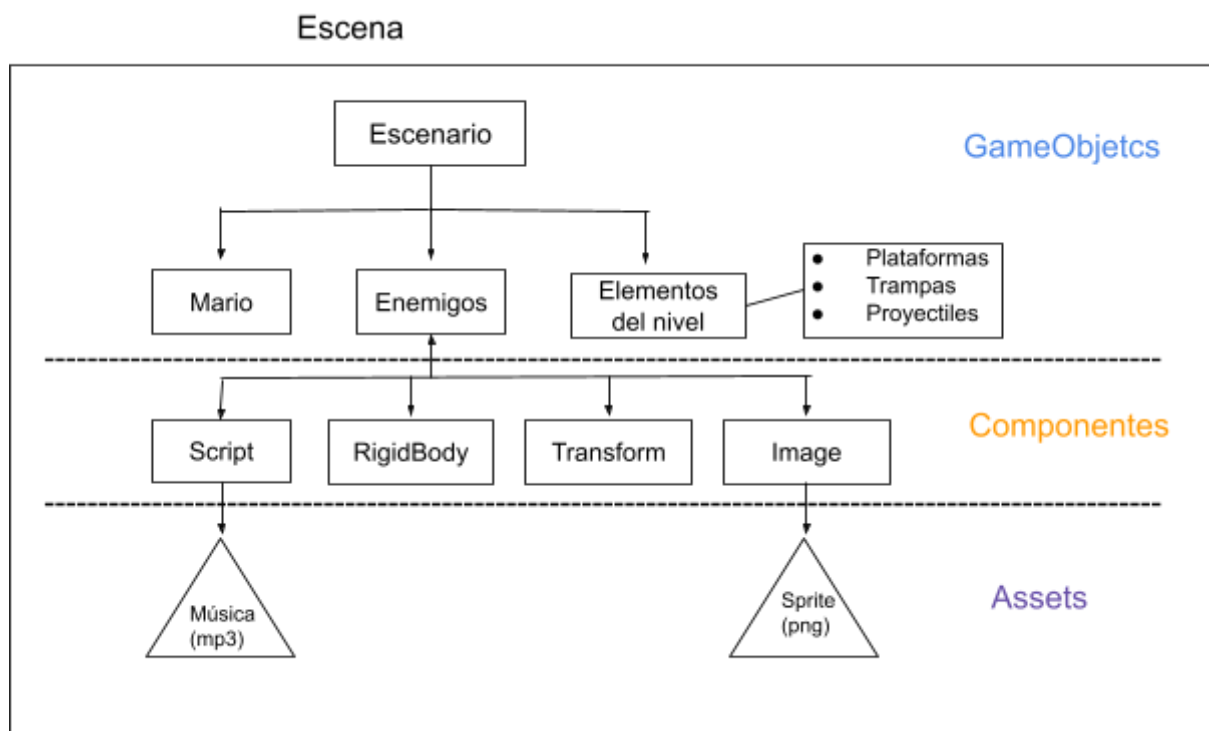


Figura 4.2.1 Organización de los elementos básicos de Unity en una escena. Diagrama interno

Se representa datos por medio de tablas relacionadas. Las filas se llaman tuplas y las columnas variables. En este caso, exceptuando al jugador (Player) se usan nombres en español para estar acorde con los scripts del videojuego en Unity, ya que los nombres de clases y métodos y variables se encuentran en este idioma.

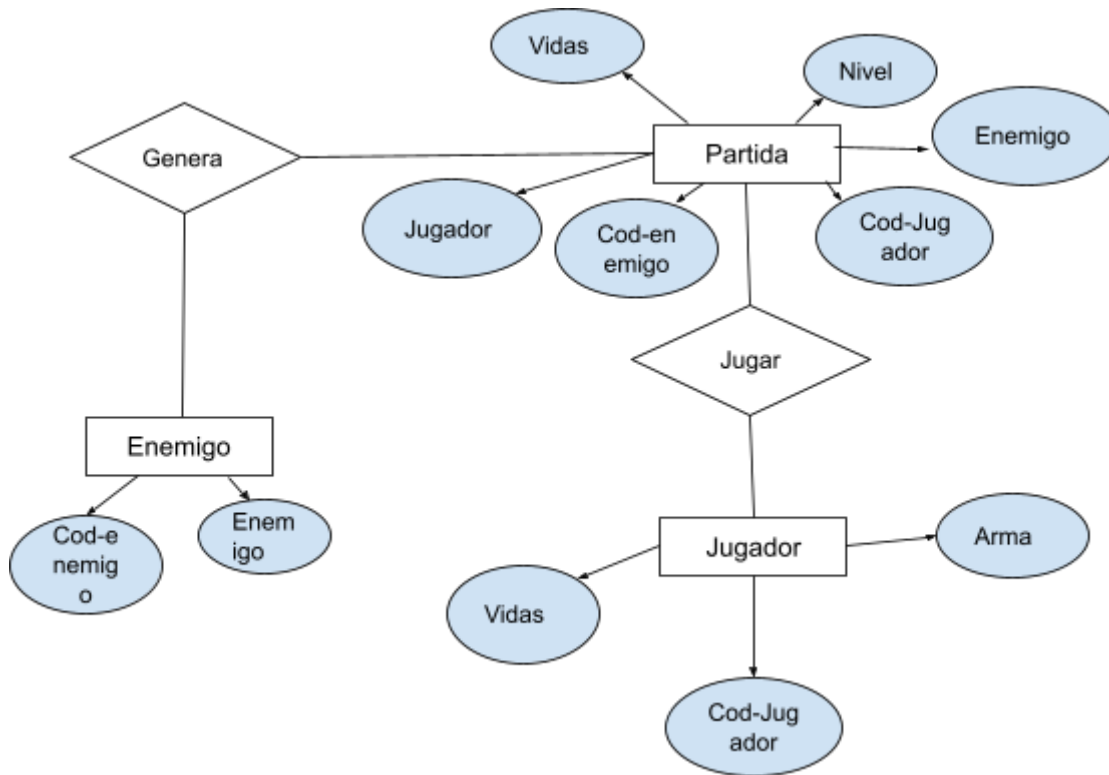


Figura 4.2.2 Diseño Conceptual. Modelo Entidad-Relación.

El videojuego tendrá un menú de navegación previo a la pantalla de juego. Este permite acceder al juego o salir de él. Se comenzará con este menú una vez que se inicie la aplicación.

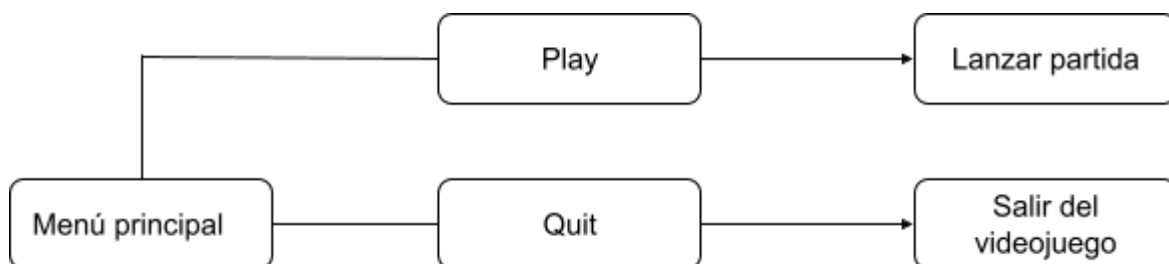


Figura 4.2.3 Diagrama del mapa de navegación..

Contenidos incluidos en la aplicación:

*Música:* está se ha sacado de una página “jsfxr” donde se ha elegido una acorde al tipo de jugao a elaborar, por lo tanto se ha escogido una música en 8 bit y en formato mp3.

*Efectos de sonido:* estos se han sacado y elaborado de una forma exclusiva y única en la página (poner página y explicar en qué consiste la página)

Para implementar la música en nuestro juego se ha usado la herramienta Audio Source en Unity, dividido en dos pistas de sonidos, la música y los efectos para así poder controlar el volumen.

Una vez hecho esto y con la ayuda de unos scripts se ha podido controlar el momento en el que empieza a sonar la música y los efectos. A continuación se detalla la lista de los efectos que se incluyen en el videojuego:

- Efecto de sonido de salto (jump.wav)
- Efecto de sonido al coger la vida (pickupCoin.wav)
- Efecto de sonido al sufrir daño el jugador (hitHurt.wav)
- Efecto de sonido del jugador al ser golpeado
- Efecto de sonido al pasar de nivel (victoriaNivel.wav)
- Efecto de sonido al morir el personaje (gameOver.wav)

*Tilesets:* este conjunto de texturas, en formato png, reunidas en una misma imagen, es decir, una composición. La implementación en el juego para elaborar el escenario de cada nivel: suelos, techos, paredes, ascensores, plataformas, etc. Para crear este conjunto se ha usado la webapp “Piskel”, donde mediante una cuadrícula compuesta por cuadrados del mismo tamaño se ha fabricado estos elementos, para el posterior aplique en la escena dentro de Unity.

*Sprites:* representan un personaje o un objeto y se utilizan para que se puedan mover de forma independiente con respecto al fondo. Para crear estos se ha utilizado “Piskel” y en formato png como forma de exportación. Por otro lado los sprites para los botones del menú de inicio y el de pause se han creado en la página “flamingtext”.

---

### 4.3. Diseño gráfico e interfaces

#### Guión

Cuando se pensó en la idea de desarrollar “Mario Vaquerizo perdió a Alaska Chaqueta”, se pusieron varios objetivos iniciales muy importantes que ayudarán a sentar unas bases que facilitarán su desarrollo y fueron la narrativa y la ambientación. Con una historia argumental, crear los diseños conceptuales del proyecto resultó mucho más fácil, especialmente tratándose de la creación de escenarios. Dado que ya había un guión argumental al que ceñirse, era más fácil identificar los lugares más importantes de la narrativa y se pudo centrar más la atención en ellos. Por otro lado, se planteó la posibilidad de utilizar realmente un referente literario directo para el proyecto, utilizando su historia como base para crear directamente los diseños; lamentablemente, aunque visualmente se podía añadir un estilo personal. Para este proyecto, además de aplicar todo lo aprendido en la investigación teórica sobre el concept art, se quería aportar un toque completamente personal y único en todos los aspectos, exigiendo así un reto mucho más complejo pero fructífero para el aprendizaje.

Para desarrollar el hilo argumental de este trabajo, se tomaron algunas decisiones importantes: Para empezar se decidió que el “guión” fuera corto ya que este proyecto se trata de una idea en proceso que sirviera como puente para desarrollar en práctica todo lo aprendido durante el estudio teórico, por lo que hacer una historia muy extensa desde el principio podría crear muchos inconvenientes e incoherencias en el desarrollo. Tomada esa decisión, se procedió a decidir la ambientación y el tipo de mundo que se quería mostrar, usando las referencias literarias y de videojuegos mencionadas anteriormente como fuente de inspiración. El juego estaría basado en una época similar a la actual, ya que en cada pantalla se encuentra un sitio famoso culturalmente de la ciudad donde se ubica el nivel. Este será el lugar de nacimiento del boss o jefe de cada escenario. Se han incluido estos personajes dentro del guión para que al diseñar los escenarios, estos cobrarán cierto sentido con su inclusión. Se tratan de Mario Vaquerizo, el protagonista y aspirante a héroe si consigue rescatar a Alaska de garras de sus enemigos, Raphael, jefe de Jaen (escenario nivel 1), Las veneno, jefas de Móstoles (escenario nivel 2), Chikilicuatre, jefe de Barcelona (escenario nivel 3), Rosa de España jefa de Granada(escenario final) .

En un mundo en el que la injusticia de Eurovisión rebosa por todos los poros de la tierra y está llena de malos recuerdos para los españoles y para los cantantes que han representado a España en las ediciones anteriores. Alaska ha ganado el festival de eurovisión y a pesar de ser laureada por muchos españoles gracias a su triunfo, no todo el mundo está contento con esa decisión y es por eso que el comité de los representantes de eurovisión de ediciones anteriores, encabezado por Rosa López o también llamada “Rosa de España” ha raptado a Alaska, porque piensan que no es merecedora de ese título.

Debido a esto Mario Vaquerizo tendrá que enfrentarse a los representantes, es decir, a los jefes de cada nivel y derrotarlos uno a uno para poder salvar a Alaska.

Story-boards (Nivel 1):

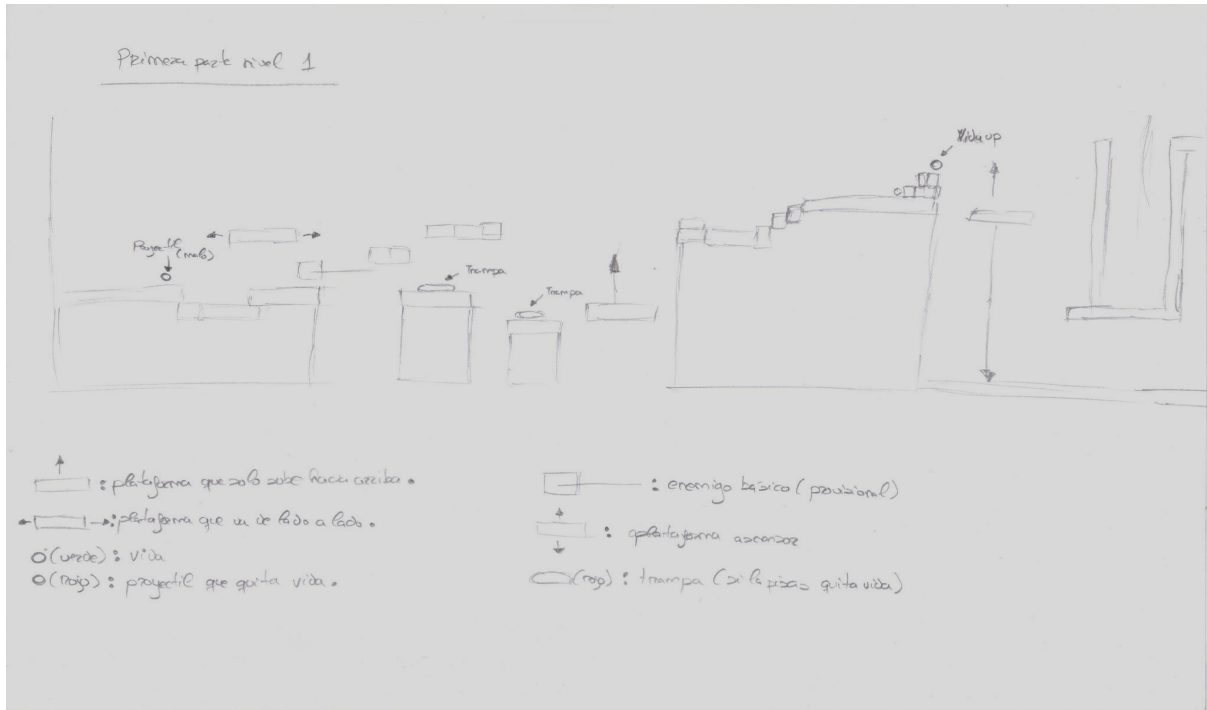


Figura 4.3.1 Storyboard nivel 1 (parte 1).

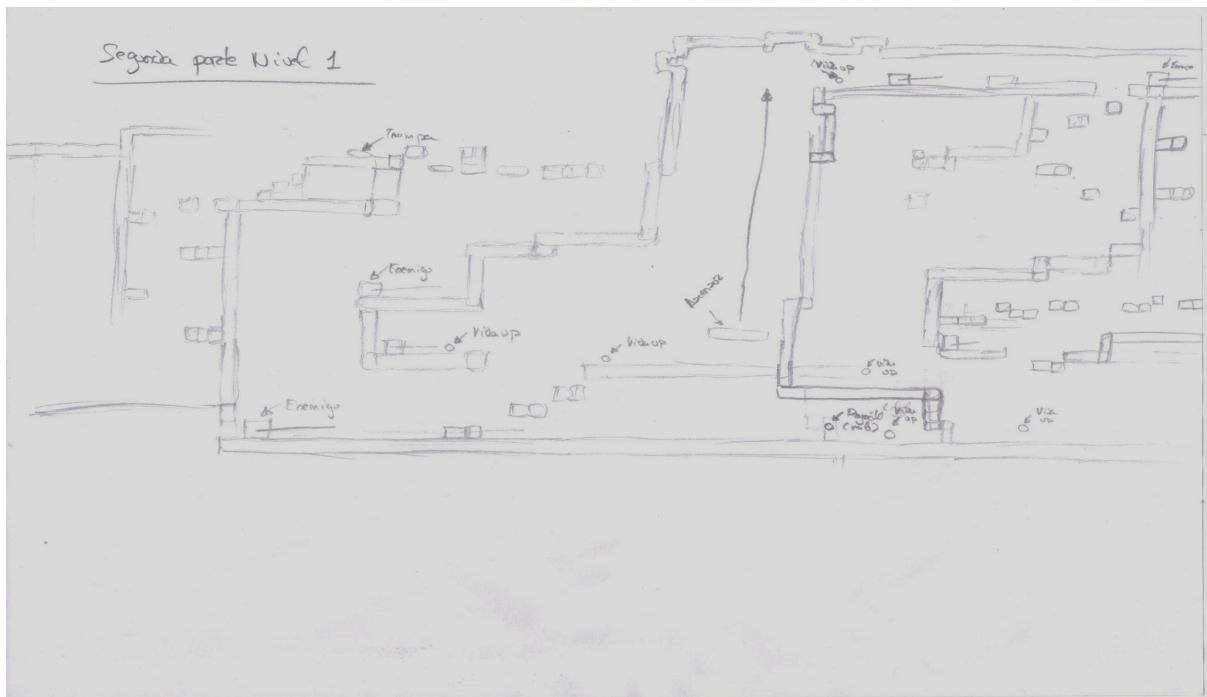


Figura 4.3.2 Storyboard nivel 1 (parte 2).

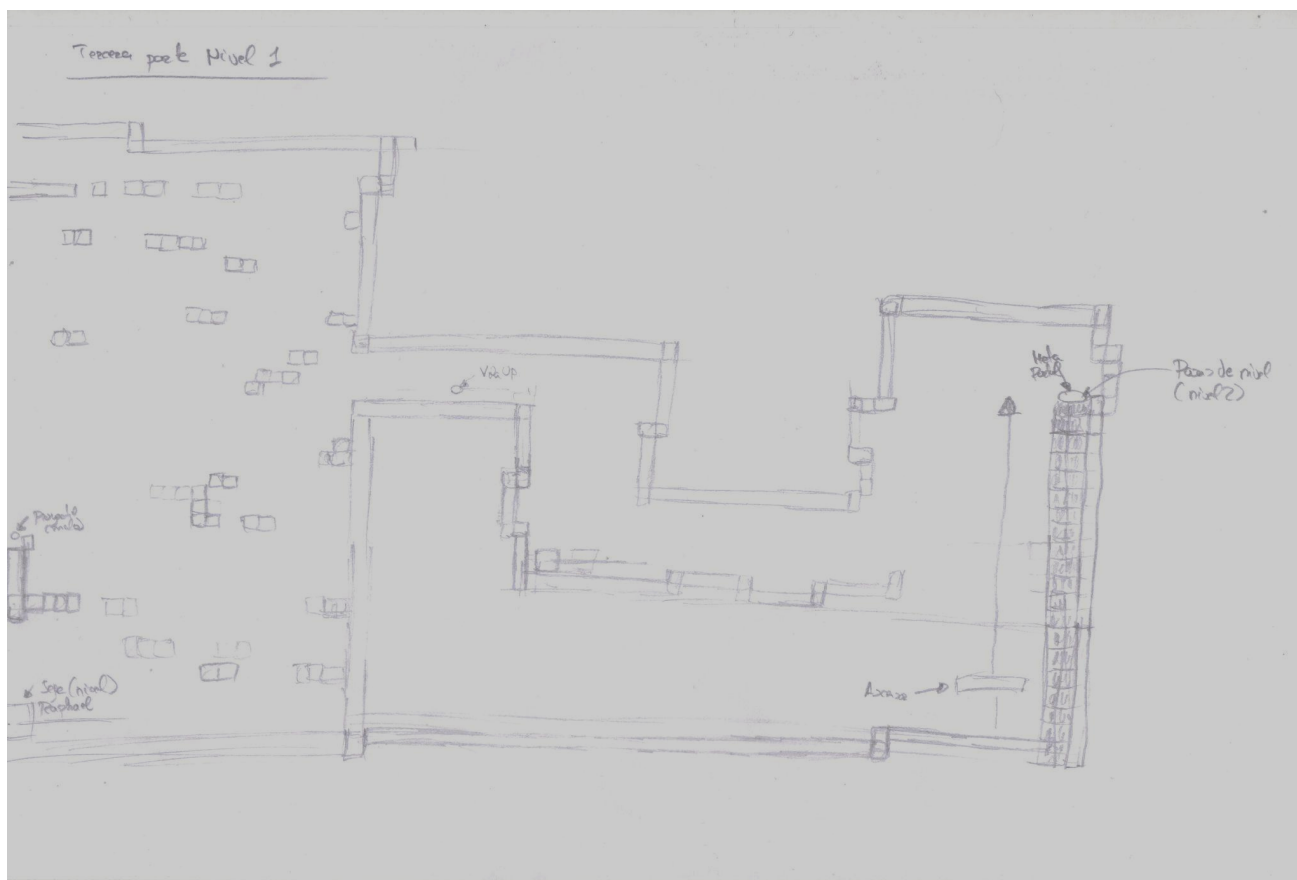


Figura 4.3.3 Storyboard nivel 1 (parte 3).

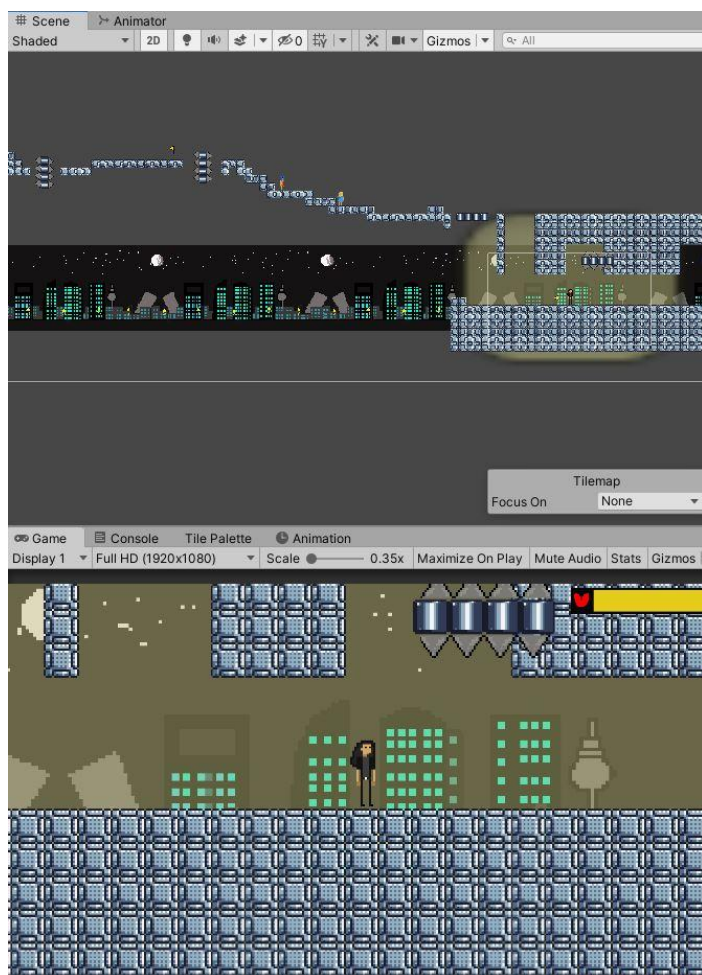


Figura 4.3.4 Implementación del diseño del nivel 1 en Unity.



Bocetos de los personajes:

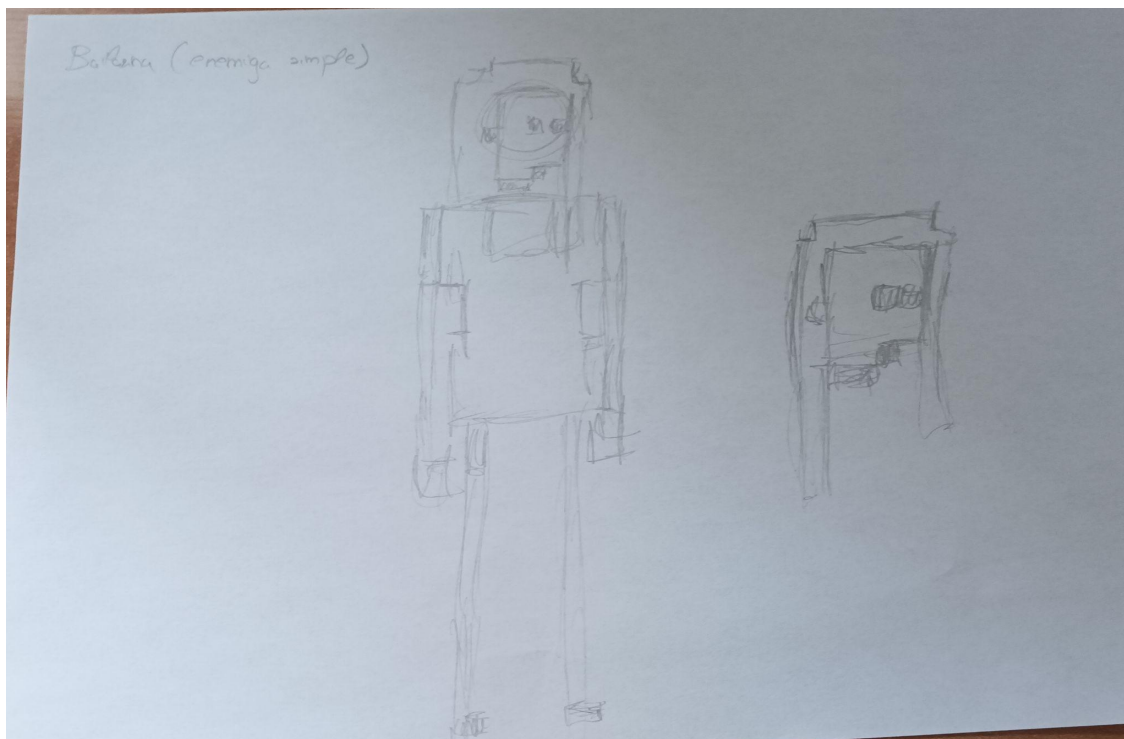


Figura 4.3.5. Boceto del personaje Mario Vaquerizo (jugador).

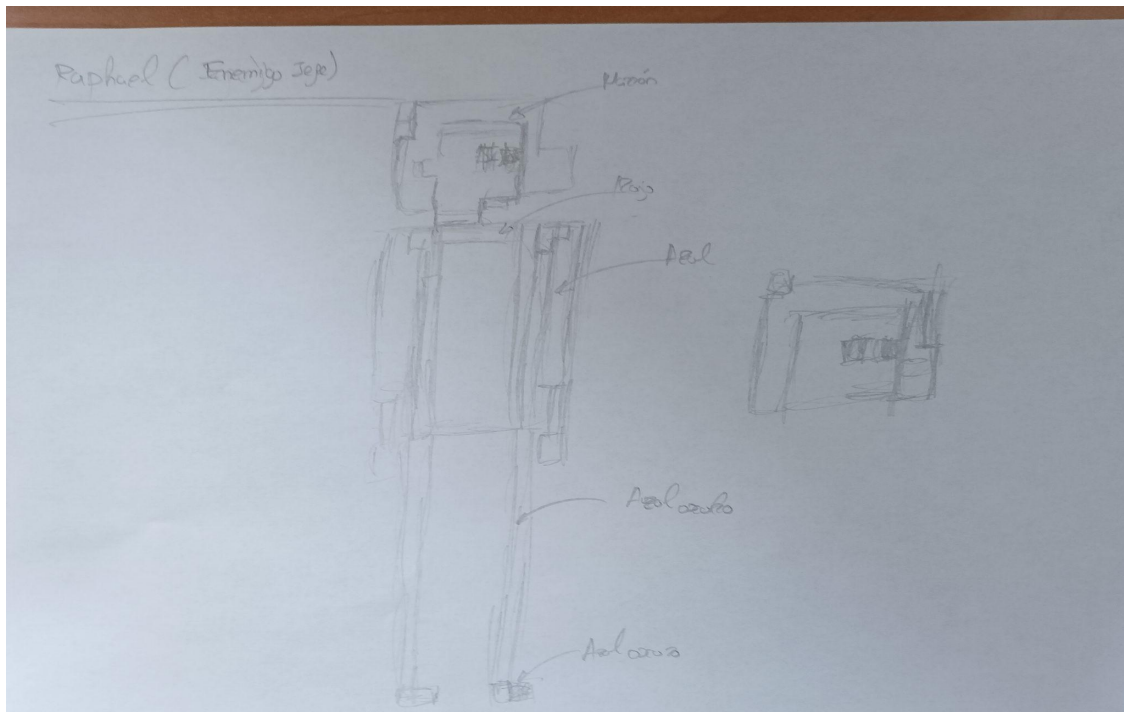


Figura 4.3.6. Boceto del personaje Raphael (enemigo jefe).

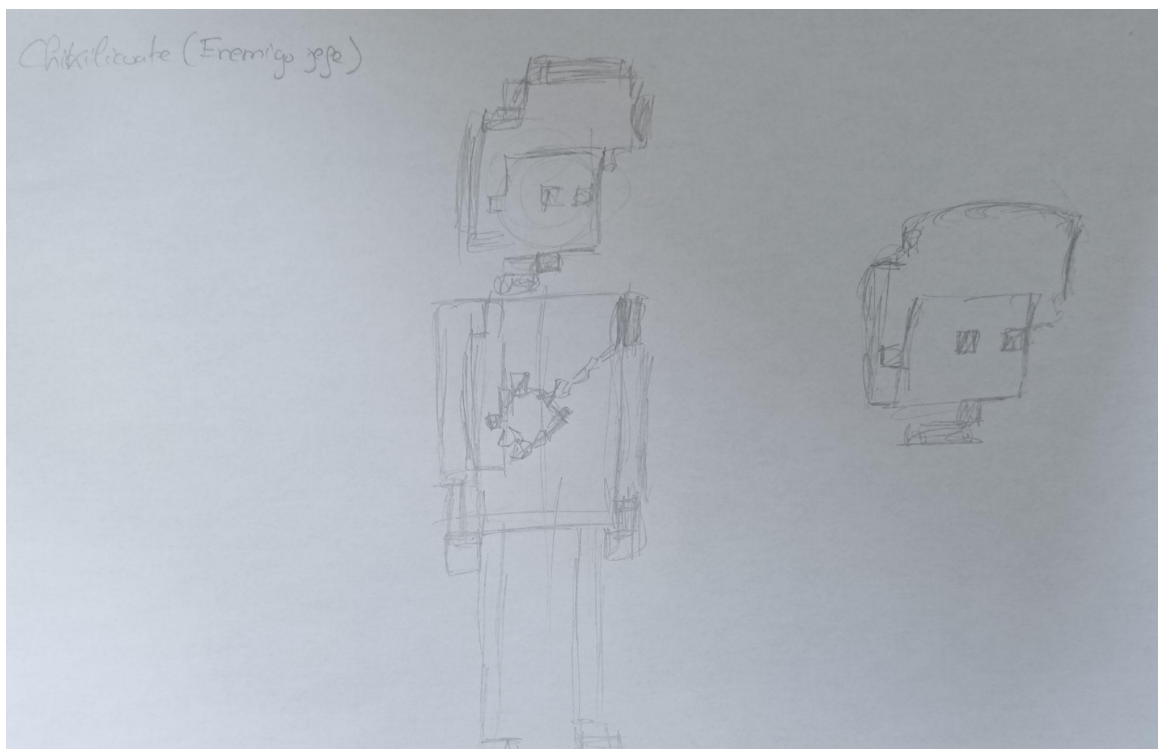


Figura 4.3.7. Boceto del personaje Chikilicuatre (enemigo jefe).

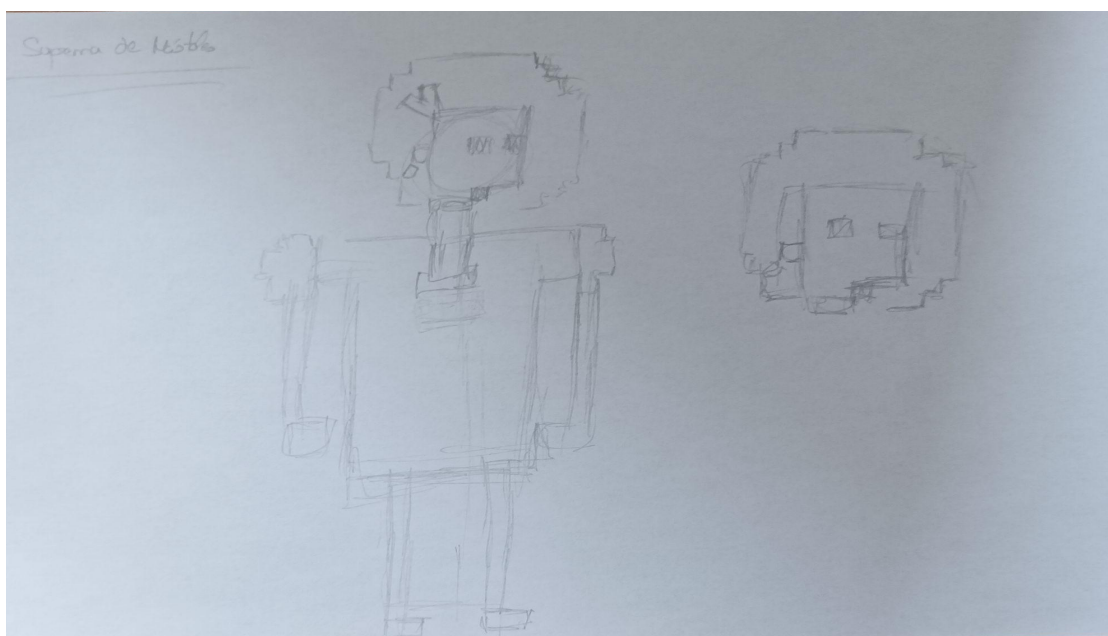


Figura 4.3.7. Boceto del personaje Suprema de Móstoles (enemigo cañón).

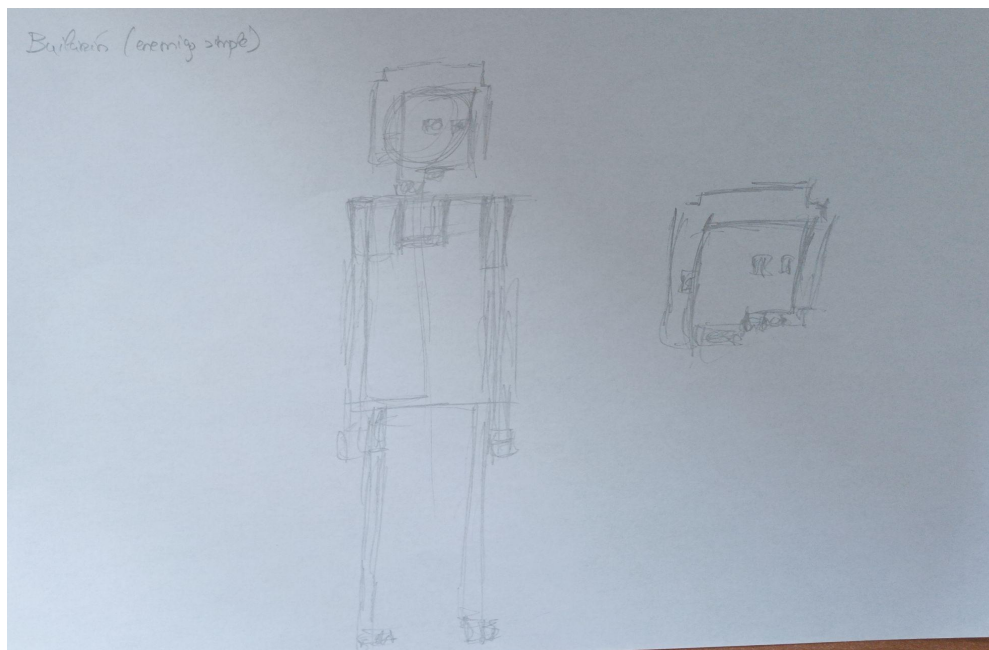


Figura 4.3.8. Boceto del personaje Bailarín (enemigo simple).

Personajes:



Figura 4.3.7. Diseño final de Mario Vaquerizo (jugador).



Figura 4.3.8. Raphael (enemigo jefe).



Figura 4.3.9. Chikilicuatre (enemigo jefe)



Figura 4.3.10. Rosa (enemigo jefe).



Figura 4.3.11. Suprema (enemigo cañón) Figura 4.3.12. Bailarín (enemigo simple)



Figura 4.3.13. Bailarina (enemigo simple).



Figura 4.3.14. Alaska.

Elementos del nivel:



Figura 4.3.16. Cerveza (aumento de vida y arma de Mario).



Figura 4.3.17. Pincho.



Figura 4.3.19. Checkpoint.



Figura 4.3.20. Cambiar de nivel.

Fondo del nivel:

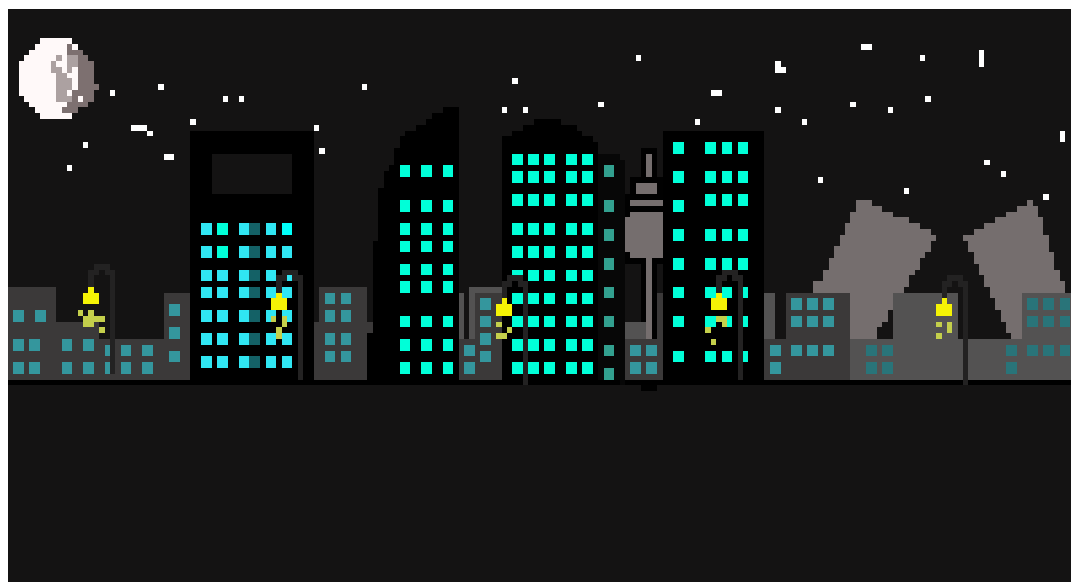


Figura 4.3.21. Fondo del nivel.

### 4.3.1. Estilos

#### Logotipo

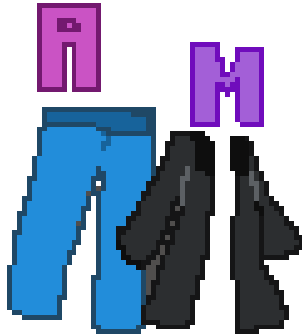


Figura 4.3.1.1. Logotipo del videojuego.

#### Anagramas

Mario Vaquerizo perdió su Chaqueta.  
El vaquero de Mario no encuentra la Chaqueta.  
El vaquero Mario no se ha puesto la Chaqueta.

#### Estilo gráfico

Antes de empezar a desarrollar cualquier diseño, primero era de vital importancia tomar una decisión y esa era la de definir el estilo gráfico. En un principio se pensó en la posibilidad usar un estilo cartoon, inspirado por juegos como Super Mario Bros o Donkey Kong. Pero tras seguir investigando distintos referentes y descubrir los estilos visuales de videojuegos muy conocidos como Diseño y Concept Art de un Videojuego: Escenarios. Chad y Jared Moldenhouer (Cuphead), Arndt Jensen (Limbo). Debido a estas influencias se decidió emplear un tipo de dibujo más tipo “cartoon”. La razón por la que se tomó esta decisión es porque se buscaba un tipo de estilo en el que aun solo se tratara de unos diseños de escenarios, estos presentaran una cierta personalidad por sí mismas, llenos de peculiaridades que los hicieran únicos. También permite la posibilidad de trabajar ciertas tonalidades de colores que muy probablemente en un estilo más realista no habrían funcionado. Con ello, todos los artistas utilizados como fuente de inspiración, especialmente los que han trabajado desarrollando el videojuego Hollow Knight, Ari Gibson y William Pellen, han servido de gran referencia a la hora desarrollar el estilo.

La paleta de colores usada en el videojuego cambia en cada nivel, tanto el fondo como el estilo de los tilesets, empleados para limitar y para los elementos en escena, produciendo una riqueza en ellos propia de cada escenario.

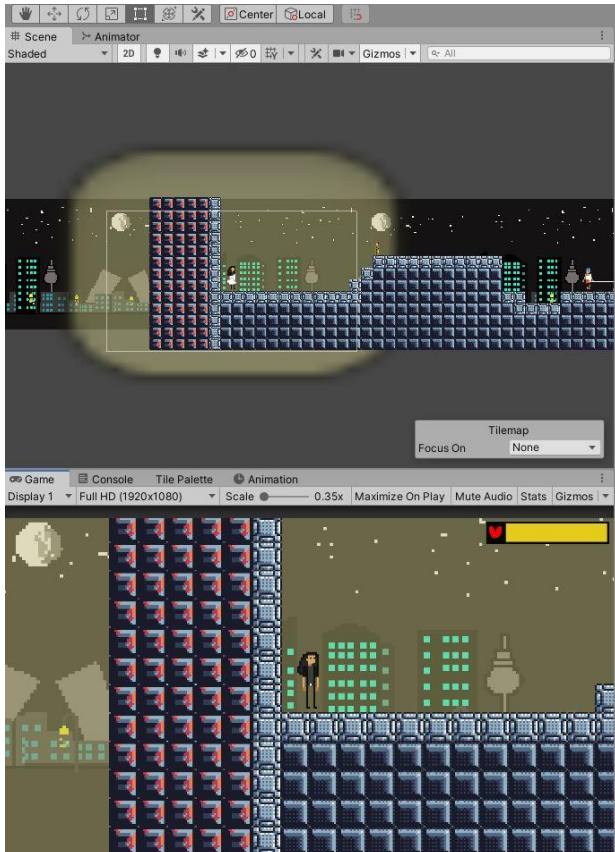


Figura 4.3.1.2. Nivel 1

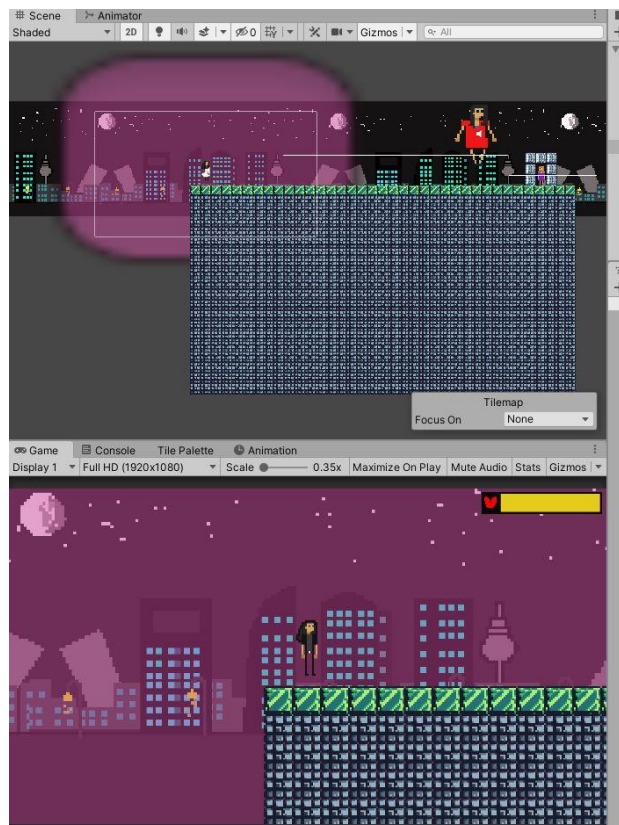


Figura 4.3.1.2. Nivel 2

### 4.3.2. Usabilidad /UX

Mario Vaquerizo (jugador) tendrá una serie de mecánicas para pasarse la pantalla:

- Movimiento del jugador: se realizará mediante las teclas A y D. A para ir hacia la izquierda y D para ir hacia la derecha.
- Salto del jugador: se realizará pulsando la tecla W.

Enemigos:

- Bola de cañón: puede disparar en un ratio en concreto. Se puede girar para seguir golpeando al jugador. Aunque estos proyectiles podrán ser esquivados por el jugador.
- Enemigo simple: se mueve de izquierda a derecha, estableciendo una zona de movimiento. Al impactar con el jugador le hace daño. Para acabar con ellos el jugador tendrá que saltar encima suya.
- Enemigo complejo (jefe del nivel): igual que el enemigo simple, pero con la diferencia que el jugador tendrá que saltar encima más veces, entre 3 y 5 veces.

Cuando la experiencia del jugador consiga o supere la experiencia requerida para subir de nivel, se alcanzará un nuevo nivel. Es decir, cuando se derrote al jefe del nivel. Conforme se vaya avanzando de niveles los enemigos simples y el enemigo final de cada nivel serán más difíciles de derrotar.

El juego contará con varios menús, todos estos menús detienen el transcurso del juego, los menús son los siguientes:

- Menú pausa: congela el juego pulsando una tecla (v) y desde el cual se puede continuar jugando si se da al play o salir del menú de pausa dando a quit.
- Menú principal: antes de ejecutarse el juego. Tiene las opciones de jugar al juego (play) o salir de el (quit).



Figura 4.3.2.1. Menú principal del videojuego.



Figura 4.3.2.2. Menú pausa del videojuego.

#### 4.4. Lenguajes de programación y APIS utilizadas

Estudiando las ventajas y las desventajas de cada motor para crear videojuegos y viendo que el proyecto que se quiere crear es un videojuego de plataformas 2D se ha descartado Unreal Engine, ya que está orientado a grupos de desarrolladores más grandes y no será necesario una calidad gráfica tan alta y precisa, además su aprendizaje es más complicado. Aunque sus físicas son



excelentes, es gratis al principio y se puede modelar personajes en el mismo motor, se ha decidido descartar este motor, ya que las desventajas son mayores.

GameMaker Studio 2, también se ha descartado, aunque sea idóneo para crear un videojuego 2D, pero no se podrá exportar a PC como lo hace Unity, siendo este un programa mucho más completo que permite exportar a múltiples plataformas (Windows, Android, Linux, Playstation..) y que además tiene mucha información y documentación, ya que es el más antiguo de todos estos.

Unity nos permite trabajar con dos lenguajes de programación, Javascript y C#. Se ha elegido C# en lugar de Javascript por la razón de que la comunidad de usuarios de Unity que utilizan el lenguaje C# es mayor y es un plus a la hora de encontrar respuestas a posibles problemas.

A parte de C# es un lenguaje orientado a objetos muy parecido a otros lenguajes vistos durante la carrera. Lo que facilita la comprensión y el aprendizaje.

#### Monodevelop

Es un entorno de desarrollo integrado, libre y gratuito y diseñado principalmente para lenguajes como C#. Monodevelop es un IDE multiplataforma que cuenta con soporte para GNU, Linux, Windows y Mac.

#### Visual Studio Code 2019

Es un Integrated Development Environment (IDE) que se ha utilizado para crear los scripts y clases (en C#) que implementa la funcionalidad del juego.

## 5. Implementación

### 5.1. Requisitos de instalación

Para realizar este proyecto solo se ha necesitado instalar el programa Unity, motor para crear el videojuego, porque para la parte gráfica he usado la webapp "Piskel".

Los requisitos de Unity dependen mucho de cada proyecto, pero para lo que se ha trabajado en este TFG no se requiere tener un ordenador potente.

Esta instalación se divide en dos: los requisitos de Unity Hub (instalador) y los requisitos para desarrollar el videojuego en Unity.

| REQUISITOS DE UNITY HUB  |   |
|--------------------------|---|
| <b>Sistema operativo</b> | Windows 7 sp+1, 8, 10 pero solo en sus versiones de 64 bit, Max OS X 10.2 o superior, Ubuntu 16.04 o 18.04, CentOS 7. |
| <b>Tarjeta gráfica</b>   | Capaz de correr Dx 10   |

| REQUISITOS PARA DESARROLLAR EL VIDEOJUEGO |   |   |
|---|---|---|
|   | <i>Requisitos minimos</i>   | <i>Requisitos recomendados</i>  |
| <b>Sistema operativo</b>                  | Windows 7 (sp+1), windows 10, versiones de 64 bit.  | Windows 11 64 bit   |
| <b>CPU</b>                                | Arquitectura x64 con instrucciones sse2 (Intel Core 2 Duo, Extreme o Quad, Intel Core i3, AMD Sempron, Phenom I y II y Athlon II) | Arquitectura x64 con instrucciones sse2 (Intel Core i5, i7, AMD FX y Ryzen) |
| <b>API de gráficos</b>                    | DX 10 y 11  | DX 12   |
| <b>Requisitos adicionales</b>             | Drivers de hardware compatibles   | Drivers de hardware compatibles   |
| <b>Gráficos</b>                           | Tarjeta de vídeo de 512 mb (GTX 650 en adelante)  | RTX 3070 (para proyectos 3D)  |
| <b>RAM</b>                                | Desde 4Gb   | 16Gb  |
| <b>Espacio en disco</b>                   | 10Gb  | 10Gb  |

## 5.2. Instrucciones de instalación

Debido a que el juego se ha desarrollado en Unity es necesario instalar dicho software. En la página oficial de Unity, se puede descargar gratuitamente; [unity3d.com/download](https://unity3d.com/download).

Una vez descargado el software de instalación, "Unity Hub". Es decir, se descarga primero un administrador de las versiones de Unity para poder instalar el programa finalmente. Además de servir como instalador, también permite manejar las opciones de crear un proyecto o acceder a un trabajo previamente guardado, es decir, es una plataforma que actúa como una nube donde puedes ver y crear nuevos proyectos.

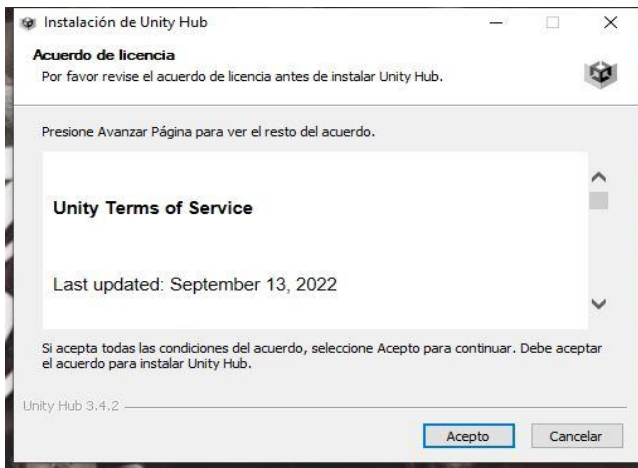


Figura 5.2.1. Instalación Unity Hub.

Estos proyectos se dividen en 2D o 3D. La diferencia es la profundidad, ya que en los juegos 3D se utiliza la dimensión z (perspectiva 3D), en cambio en los juegos 2D solo se utiliza el eje “x” y el eje “y”.

Siempre que se lanza el editor de Unity (unity hub), se muestra en la Pantalla de Inicio.

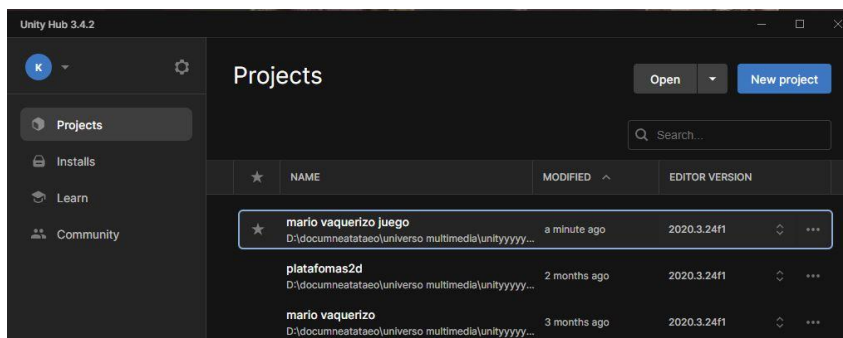


Figura 5.2.2. Instalación Unity Hub.

En la esquina superior derecha se puede crear un nuevo proyecto.

Se puede elegir entre las diferentes opciones, dependiendo a lo que se quiera orientar dicho proyecto. Hay algunas opciones que se necesita descargar un template antes de poder crearlo. Sin embargo, para realizar este videojuego no se ha precisado descargar nada.

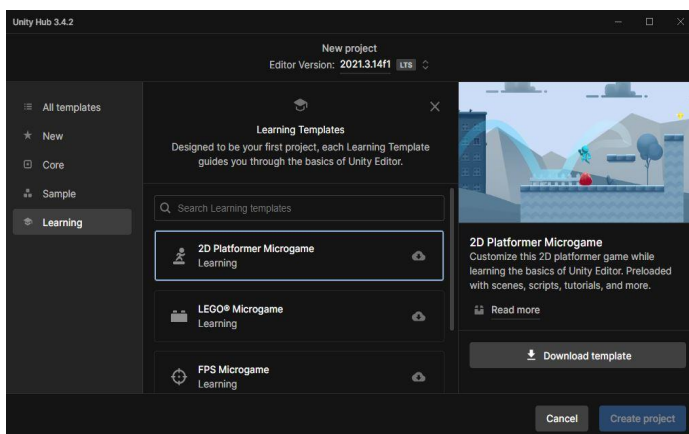


Figura 5.2.3. 2D Platform Microgame.

Con el template vacío 2D core bastará. Lo escogemos. Después se decidirá una ruta para ubicar el proyecto nuevo.

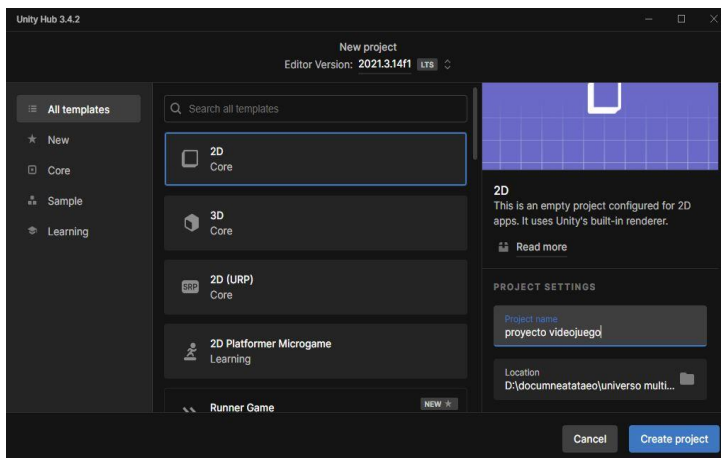


Figura 5.2.4. 2D Core.

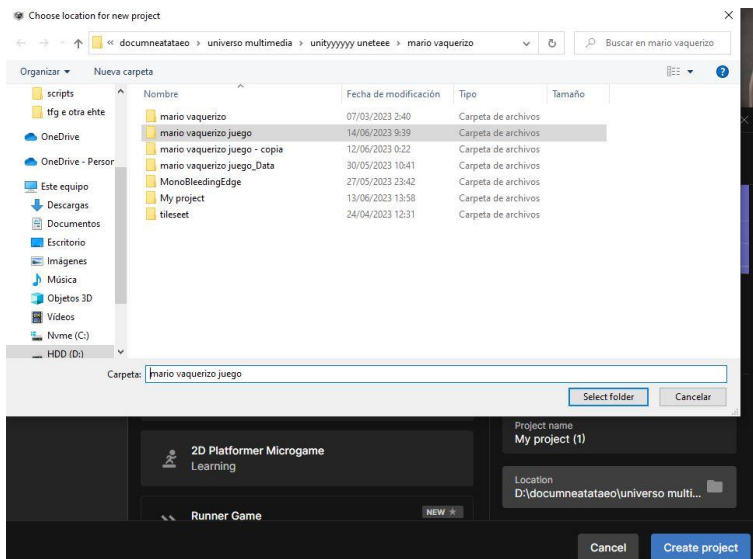


Figura 5.2.5. Ruta para ubicar el proyecto nuevo.



Figura 5.2.6. Pantalla de carga del nuevo proyecto.

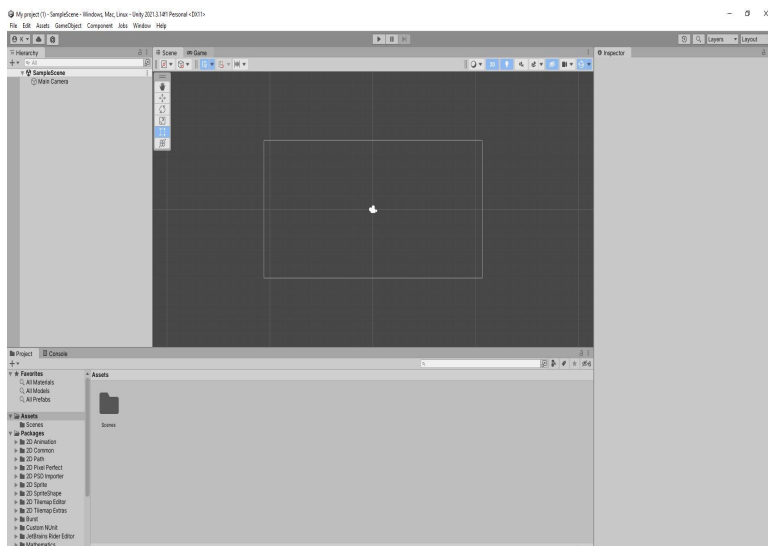


Figura 5.2.7. Proyecto nuevo una vez cargado.

## 6. Implementación

### 6.1. Instrucciones de uso

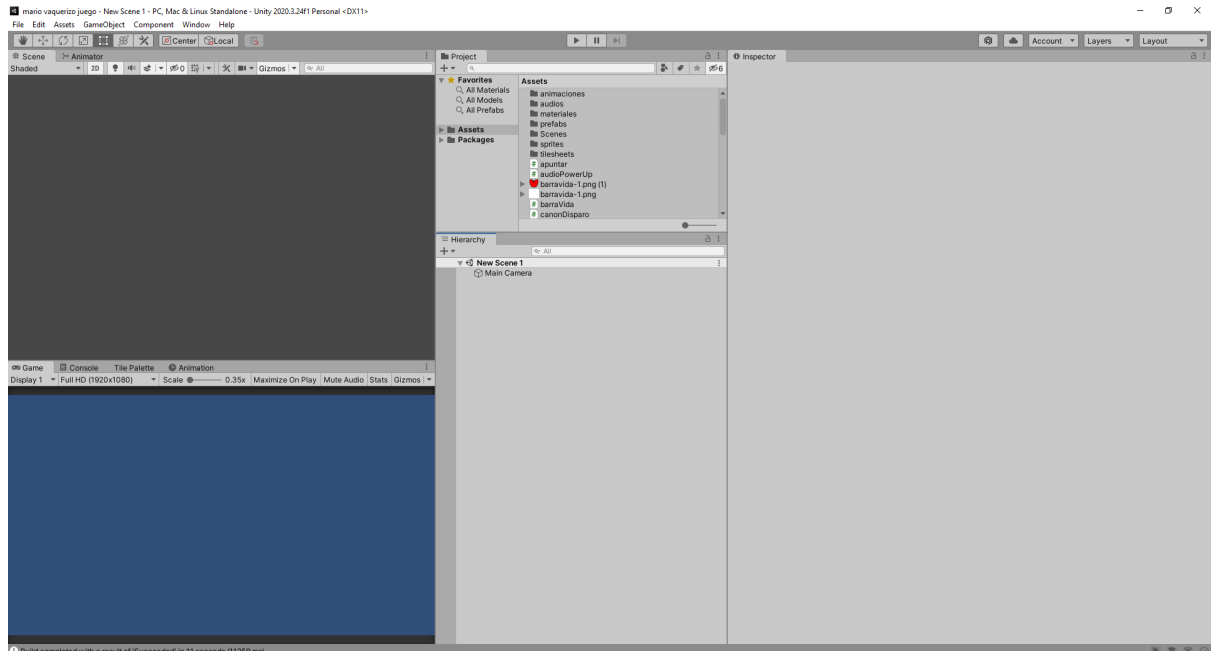


Figura 6.1.1. Proyecto nuevo una vez cargado.

## Paneles:

Unity se divide en paneles y cada uno de ellos tiene una función.

### **Panel de jerarquía**

Contiene un listado de todos los de todos los objetos que se incluyen en la escena del juego.

### **Panel de escena**

Representación gráfica de todos los elementos de la escena.

### **Panel de juego (game)**

Nos muestra cómo se vería nuestro videojuego desde el punto de vista del jugador.

### **Panel inspector**

Nos detalla todas las propiedades del elemento que hemos seleccionado. En este podremos añadir componentes que se implementarán en este objeto: scripts, audio, componentes para el sprite, colliders, etc.

### **Panel de proyecto**

Es el equivalente a un explorador de ficheros para visualizar los ficheros de nuestro proyecto.

Un asset es cualquier tipo de archivo: video, audio, foto, etc.

### **Panel de consola**

Nos muestra los errores del videojuego.

## 6.2. Prototipos

### 6.2.1. Prototipos Lo-Fi

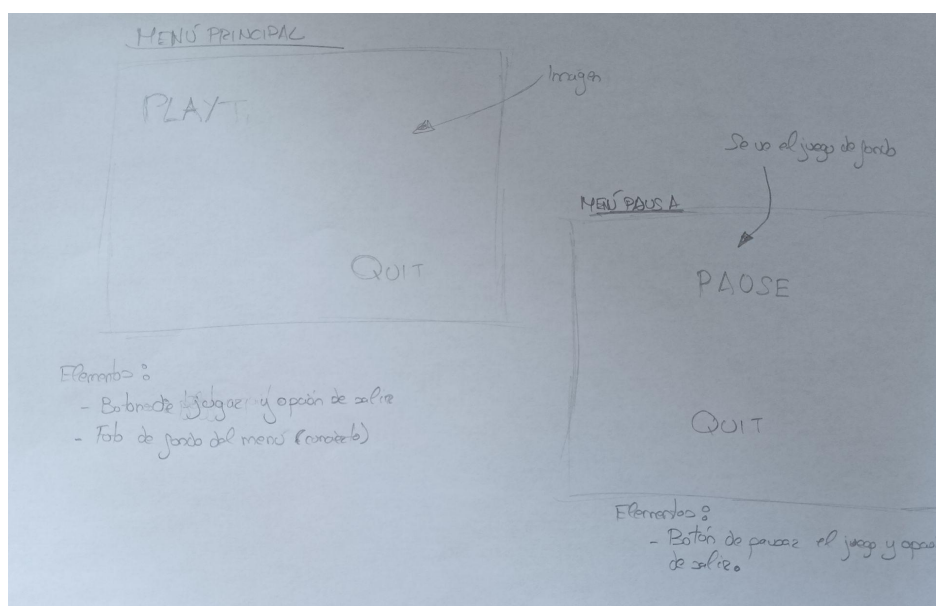


Figura 6.2.1.1. Wireframe menú principal.

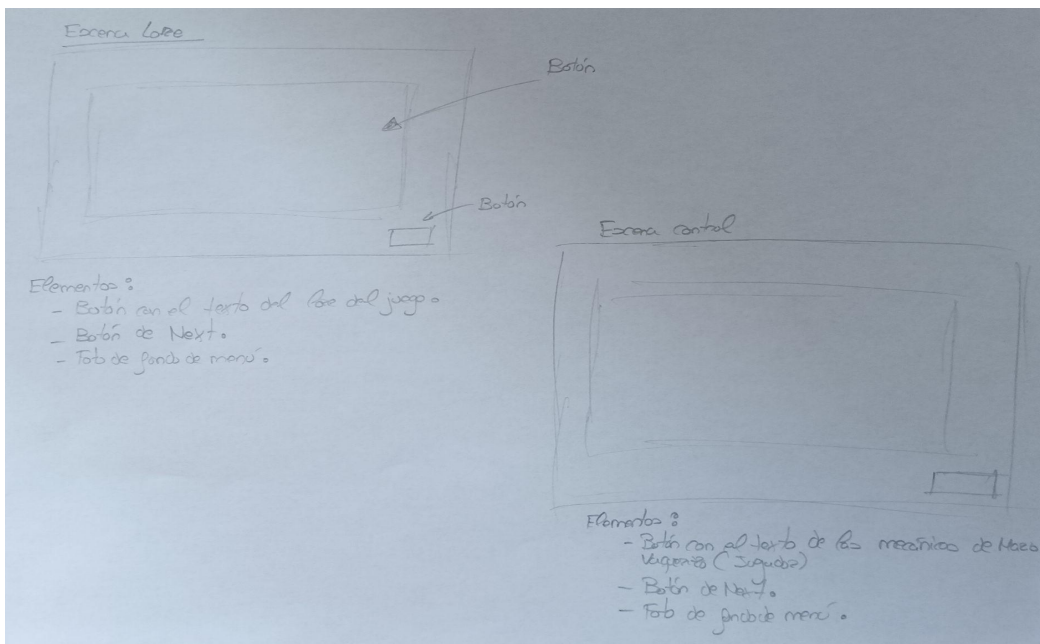


Figura 6.2.1.2. Wireframe escena Lore.

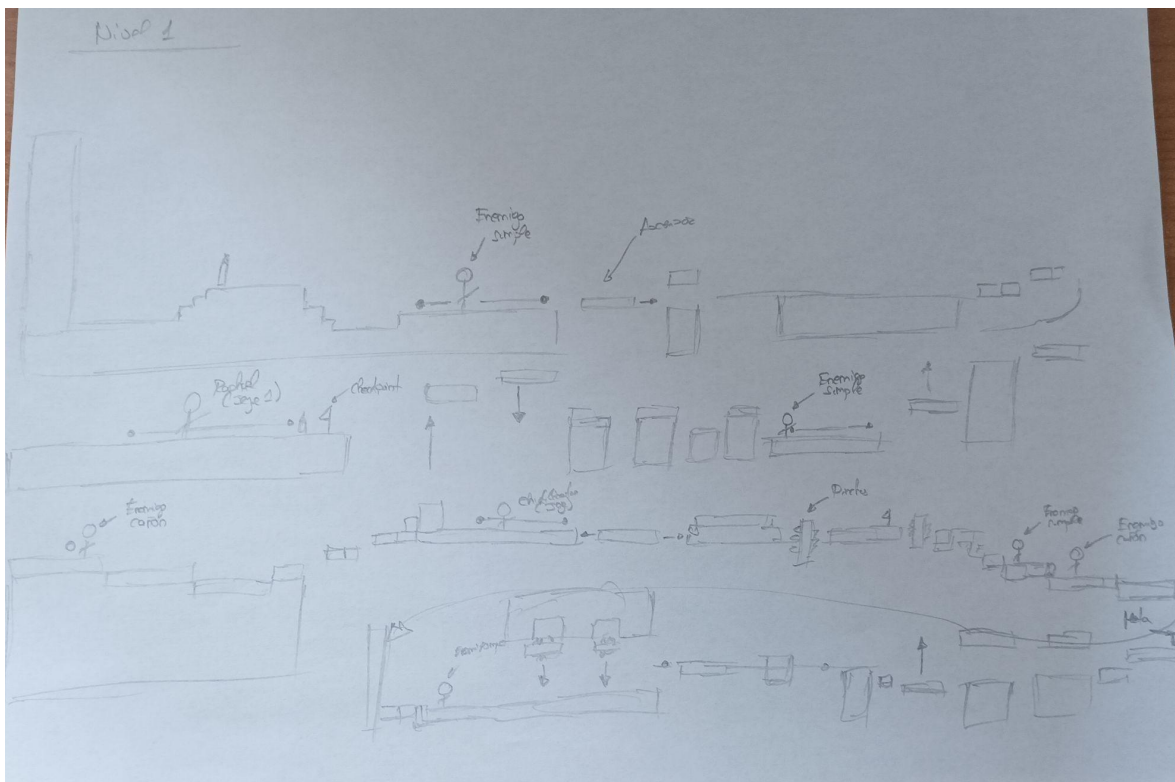


Figura 6.2.1.2. Boceto diseño final nivel 1.

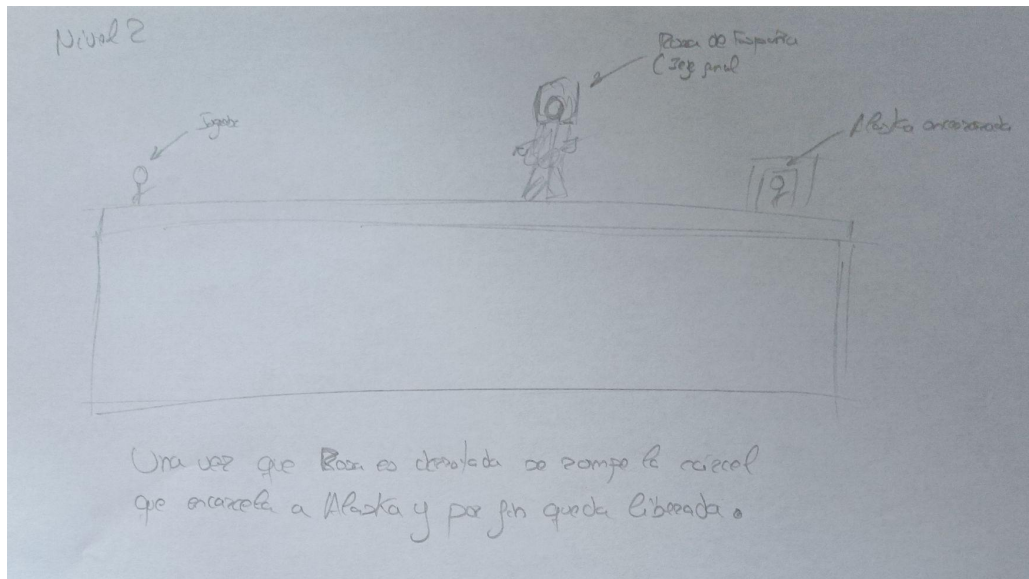


Figura 6.2.1.2. Boceto diseño final nivel 2.

## 6.2.2. Prototipos Hi-Fi

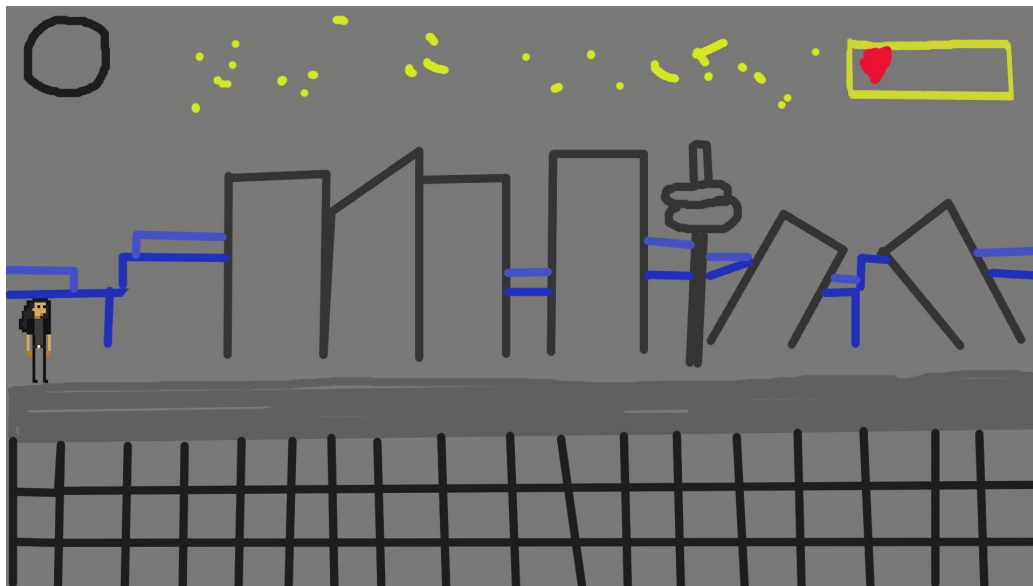


Figura 6.2.2.1. Mockup nivel 1.





Figura 6.2.2.2. Maqueta funcional menú principal.



Figura 6.2.2.3. Maqueta funcional menú pausa.



Figura 6.2.2.4. Mockup nivel 2.

### 6.3. Tests

El objetivo principal de esta evaluación es analizar la percepción del usuario sobre el juego y determinar si este proporciona a los usuarios la usabilidad necesaria para concluir con éxito su objetivo.

Lista de tareas:

- Iniciar el juego.
- Eliminar a todos los enemigos del nivel.
- Pasar al nivel 2.
- Rescatar a Alaska.

Métricas a evaluar:

1. Cuestionario de diagnóstico.

- ¿Juegas normalmente a videojuegos?
- ¿Cuántas horas a la semana dedicas a los videojuegos?
- ¿En qué dispositivos o consolas juegas?
- ¿Con quien juegas?

- 
- ¿Juegas online u offline?
  - ¿Qué género de videojuegos te gustan?

## 2.Experiencia del juego.

Sección 1: Califica con una escala del 1 al 10.

- ¿Qué tan divertido te pareció el juego? (Diversión)
- ¿Te pareció emocionante el juego? (Emoción)
- ¿Qué tan complicado fue adaptarte a la forma de control del juego? (Control)
- ¿Qué tan fácil fue cumplir con el objetivo del juego? (Dificultad)

Sección 2: contesta según la siguiente escala: Totalmente de acuerdo; De acuerdo; Neutral; En desacuerdo; Totalmente en desacuerdo.

- El juego es agradable en cuanto al control del jugador
- El juego es difícil de jugar
- Realmente sentí que estaba dentro del videojuego
- Me esforzaba para seguir jugando
- Deseaba volver a jugar
- no tuve problemas para pasar de nivel y completar los objetivos del juego

Resultados (evaluación a 4 usuarios):

## Experiencia del juego: Sección 1

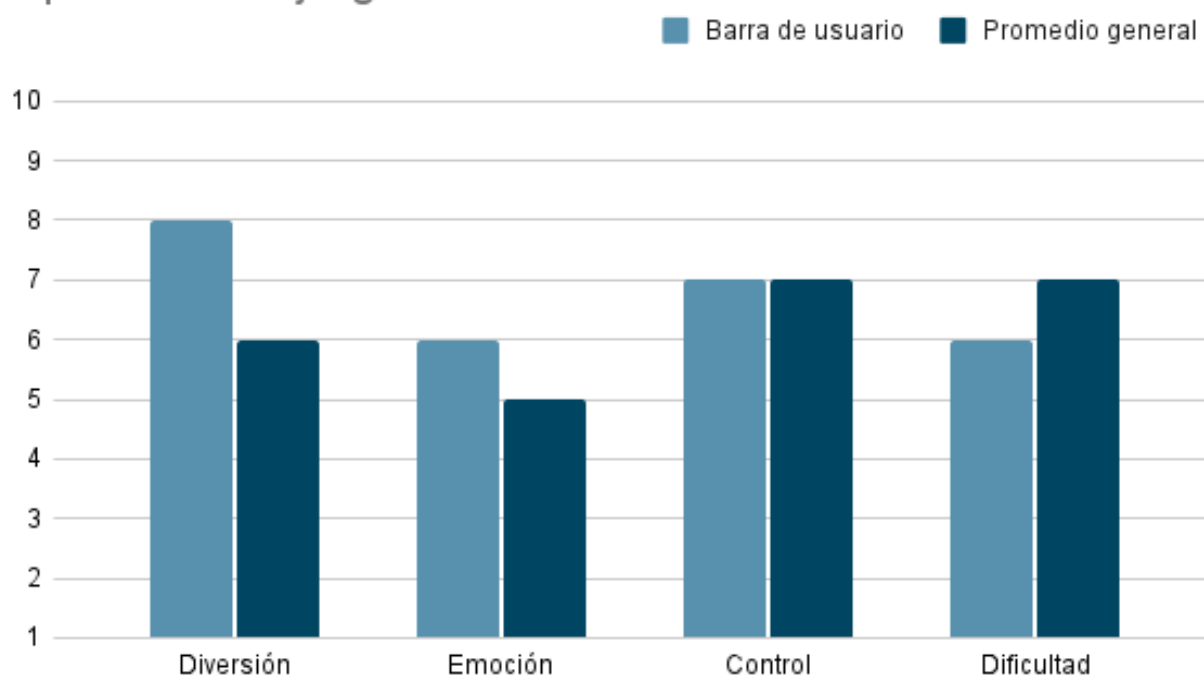



Figura 6.3.1. Resultados de la Experiencia del juego (Sección 1).


## 6.4. Guía de usuario

### MARIO VAQUERIZO (JUGADOR)



Personaje principal del juego, cuyo objetivo principal es rescatara a su amada Alaska, pero no será tan fácil, porque los enemigos y los obstáculos no se lo pondrán fácil.

MECÁNICAS:



Moveirse derecha: D y Flecha Der.


Moveirse izquierda: A y Flecha Izq.

Saltar: W, Flecha Arriba y Espacio


Disparar botellines: Botón izq. del ratón

---


### ELEMENTOS DEL NIVEL




Vater: una vez que se hayan cumplido los objetivos del nivel, el jugador podrá pasar al siguiente nivel al tocar este objeto



Checkpoint: una vez alcanzado este objeto cuando el jugador muera hara su respawn desde este punto.




Pincho: este elemento se encontrará en algunas de las plataformas móviles. Si el jugador es golpeado por este objeto perderá puntos de vida.




Ascensor/Plataforma: elemento móvil que permitirá al jugador desplazarse.

---


### PERSONAJES




Alaska: mujer de Mario Vaquerizo. Está esperando a ser rescatada por su amado.




Raphael: enemigo jefe que se moverá de un lado hacia otro. Tiene 5 puntos de vida. Si el jugador impacta con él sufrirá daños.




Raphael: enemigo jefe que se moverá de un lado hacia otro. Tiene 3 puntos de vida. Si el jugador impacta con él sufrirá daños




Suprema de Móstoles: enemigo que lanza unas bolas de cañon. Si esas bolas impactan en el jugador su vida bajará.



Bailarin: enemigo simple que se moverá de lado a lado y si el jugador impacta con el sufrirá daño.



Bailarina: enemigo simple que se moverá de lado a lado y si el jugador impacta con el sufrirá daño.



Rosa : enemigo jefe final que se moverá de un lado hacia otro. Tiene 20 puntos de vida. Si el jugador impacta con él sufrirá daños. Su arma son los vateres

Figura 6.4.1. Guía de usuario.

## 7. Conclusiones y líneas de futuro

Hacer este proyecto ha sido todo un reto para mí, incluso dudé en si hacerlo de una página web o de un videojuego, puesto que sabía que la página web me iba a costar menos, por el hecho de que durante toda la carrera hemos hecho varias páginas webs. Por lo tanto, sería más fácil, pero yo quería meterme en algo nuevo, en un proyecto que me atrajera y que me hiciera investigar y aprender cosas nuevas y es por eso que al final decidí hacer el TFG de un videojuego.

Una vez que tome la decisión sabía que empezaba prácticamente de 0, porque para crear un videojuego se necesitan tener en cuenta y saber utilizar ciertos programas que yo no había tocado nunca, así que me di cuenta que iba a ser complicado, porque para desarrollarlo tenía que aprender a desenvolverme en cada una de las disciplinas o al menos aprender lo básico para sacarlo adelante. Sin duda, un reto técnico muy complejo, sin embargo he descubierto que me apasiona este campo que además cambia día a día renovándose, innovando en nuevos proyectos y técnicas.

Con este proyecto he llegado a la conclusión de que para llevar a cabo cualquier idea creativa hace falta también técnica y procesos para hacerla algo real y tangible, desarrollar una idea plasmarla en cualquier medio es sin duda tarea complicada y más si se trata de un videojuego que puede llegar a convertirse en algo muy extenso y profundo..

### 7.2. Líneas de futuro

Este proyecto tiene mucho por recorrer. Tome una decisión muy ambiciosa y querái hacer muchos niveles, pero me di cuenta que no me daba tiempo para tanto, entonces reducí mucho el producto final. No obstante, preveo en un futuro seguir con el juego y ampliarlo, porque me ha divertido mucho hacerlo y quiero seguir aprendiendo más mecánicas e introducir más elementos al videojuego para hacerlo aún más atractivo.

---

# Bibliografía

**Scoalstici, C.** (2015). *Unity 2D Game Development Cookbook*.

**Williams, R.** (2001). *The Animator's Survival Kit* .

**Goldberg, E.** (2008). *Character Animation Crash Course*. Silman-James Press.

**Blair, P.** (1994). *Cartoon Animation*. Walter Poster Publishing.

**UNITY website:** <https://unity.com/es>, consultado 02/05/2023

**FLAMINGTEXT website:** <https://flamingtext.es/>, consultado 12/05/2023

**JSFXR website:** <https://sfxr.me/>, consultado 25/05/2023

**PISKEL website:** <https://www.piskelapp.com/>, consultado 27/05/2023

**GITHUB website:** <https://github.com/kikeleanimado/proyectoMarioVaquerizo>, 16/06/2023

**YOUTUBEwebsite:**

[https://www.youtube.com/watch?v=dwYP8p0yWXU&ab\\_channel=UnKikeTransmite](https://www.youtube.com/watch?v=dwYP8p0yWXU&ab_channel=UnKikeTransmite)

**YOUTUBE website:**

[https://www.youtube.com/watch?v=idfsc79v8Tk&ab\\_channel=UnKikeTransmite](https://www.youtube.com/watch?v=idfsc79v8Tk&ab_channel=UnKikeTransmite)

**WETRANSFER website:** <https://we.tl/t-bp1qwCoFeB>, 18/06/2023