Universitat Oberta de Catalunya uoc.edu

Umbrellibur
Un juego de plataformas 2D desde 0

Autor: Pablo Pérez Martínez
Máster universitario de Desarrollo de aplicaciones para dispositivos móviles
Área: Diseño y programación

Consultor: Francesc d'Assís Giralt Queralt
Profesor responsable de la asignatura: Carles Garrigues Olivella

Fecha Entrega: 05/06/2019

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada 3.0 España de Creative Commons

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Umbrellibur* *Un juego de plataformas 2D desde 0*
Nombre del autor:	*Pablo Pérez Martínez*
Nombre del consultor/a:	Francesc d'Assís Giralt Queralt
Nombre del PRA:	Carles Garrigues Olivella
Fecha de entrega:	06/2019
Titulación:	*Máster universitario de Desarrollo de aplicaciones para dispositivos móviles*
Área del trabajo final:	*Diseño y programación*
Idioma del trabajo:	*Español*
Palabras clave	*Videojuegos, aplicaciones, programación*

Resumen del trabajo:

En el siglo XXI las aplicaciones móviles están cambiando la sociedad y en particular el mundo de los videojuegos también se ha visto afectado.

Umbrellibur se presenta como un videojuego que, aparte de cumplir con el objetivo de entretenimiento, pretende introducir un objeto cotidiano como un paraguas en los videojuegos 2D de plataformas.

El objetivo de este proyecto es el diseño y desarrollo de un videojuego exportable a cualquier dispositivo (gracias a Unity) con especial énfasis en la jugabilidad en dispositivos móviles.

Abstract:

In the 21st century mobile applications are changing society and in particular the world of video games has also been affected.

Umbrellibur is presented as a videogame that, apart from fulfilling the goal of entertainment, aims to introduce an everyday object as an umbrella into 2D platform videogames.

The objective of this project is the design and development of a videogame exportable to any device (thanks to Unity) with special emphasis on the playability of mobile devices.

Índice

1. Introducción 1
1.1 Contexto y justificación del trabajo 1
1.1.1 Contexto 1
1.1.2 Justificación 2
1.1.3 Estudio de mercado 3
1.2 Objetivos del trabajo 4
1.3 Enfoque y método seguido 5
1.3.1 Tecnologías utilizadas 5
1.3.2 Diseño 6
1.3.3 Metodología 7
1.4 Planificación del trabajo 8
1.4.1 Recursos 8
1.4.2 Planificación 9
1.4.3 Estimación de costes 11
1.4.4 Análisis de riesgos 12
1.5 Breve sumario de productos obtenidos 14
1.6 Breve descripción de los siguientes capítulos de la memoria 15
2. Análisis 16
2.1 Análisis de usuarios 16
2.2 Fichas de usuario y escenarios de uso 20
2.3 Requisitos 22
2.4 Casos de uso 23
2.4.1 Descripción de los casos de uso 23
2.4.2 Diagramas de casos de uso 24
3. Diseño 25
3.1 Arquitectura 25
3.2 Mapa de navegación 26
3.3 Prototipo de baja fidelidad 27
3.4 Diseño de elementos 29
3.5 Prototipo funcional 31
3.6 Evaluación del prototipo funcional 33
4. Implementación 36
4.1 Herramientas utilizadas 36
4.2 Estructura del proyecto 37
4.3 Implementación del videojuego 38
4.3.1 Interfaz de usuario 38
4.3.2 Niveles de juego 42
4.4 Principales problemas encontrados 47
5. Pruebas 48
6. Publicación 50
6.1 Cuenta de Play Store 50
6.2 Publicación de la versión beta 50
6.3 Aplicación publicada 51
7. Conclusiones 52
7.1 Lecciones aprendidas 52
7.2 Análisis de la metodología del trabajo 52
7.3 Cumplimiento de objetivos 52

7.4 Líneas de trabajo futuro 53
8. Glosario 54
9. Bibliografía 55

Lista de figuras

Figura 1: Videojuegos de plataformas/aventuras 1
Figura 2: Videojuegos similares 3
Figura 3: Logos de las herramientas utilizadas 5
Figura 4: Captura de Umbrellibur 6
Figura 5: Cascada 7
Figura 6: Scrum 7
Figura 7: Planificación 9
Figura 8: Diagrama de Gantt 10
Figura 9: Tramo de edad 16
Figura 10: Sexo 16
Figura 11: Tiempo dedicado a videojuegos 17
Figura 12: Dedicación a juegos de plataformas 17
Figura 13: Motivos de utilización 17
Figura 14: Elementos cotidianos en videojuegos 18
Figura 15: Funcionalidad deseada 19
Figura 16: Casos de uso 24
Figura 17: Caso de uso: Jugar 24
Figura 18: Mapa de navegación 26
Figura 19: Prototipo baja fidelidad pantalla inicial 27
Figura 20: Prototipo baja fidelidad selección de nivel 27
Figura 21: Prototipo baja fidelidad opciones 28
Figura 22: Prototipo baja fidelidad nivel 28
Figura 23: Personaje principal 29
Figura 24: Enemigo Lobo 29
Figura 25: Enemigos stickman 30
Figura 26: Elementos decorativos 30
Figura 27: Elementos interactivos 30
Figura 28: Prototipo alta fidelidad pantalla inicial 31
Figura 29: Prototipo alta fidelidad selección de nivel 31
Figura 30: Prototipo alta fidelidad opciones 31
Figura 31: Prototipo alta fidelidad instrucciones 32
Figura 32: Prototipo alta fidelidad pantalla historia 32
Figura 33: Prototipo alta fidelidad nivel 32
Figura 34: Valoración de la UI 33
Figura 35: Valoración del movimiento 33
Figura 36: Valoración estética 34
Figura 37: Valoración de si es intuitivo 34
Figura 38: Valoración de uso 34
Figura 39: Valoración general 35
Figura 40: Valoración del prototipo 35
Figura 41: Unity 36
Figura 42: Visual Studio 36
Figura 43: Estructura Assets 37
Figura 44: Menú de inicio 38
Figura 45: Menú de niveles 38
Figura 46: Pantalla de logros 39
Figura 47: Menú de opciones 39

Figura 48: Instrucciones 40
Figura 49: Historia / Mensaje 41
Figura 50: Interfaz del juego 41
Figura 51: Nivel de juego 42
Figura 52: Fragmento de máquina de estados 42
Figura 53: Trampas 43
Figura 54: Estados enemigos 44
Figura 55: Enemigos 44
Figura 56: Enemigo final 44
Figura 57: Lluvia normal 45
Figura 58: Lluvia negra 45
Figura 59: Fuego 45
Figura 60: Fases creación Play Store 50
Figura 61: Consola Google Play 50
Figura 62: Ficha Play Store 51

1. Introducción

En este capítulo se desarrolla una fase inicial de planificación del proyecto del Trabajo de Fin de Máster (TFM).
En los siguientes apartados, además de planificar el desarrollo de la aplicación estimando tiempos y recursos, se analiza el contexto de la aplicación, se justifica su necesidad y se analizan los objetivos que debe cumplir la misma.

1.1 Contexto y justificación del trabajo

1.1.1 Contexto

En el siglo XXI se está produciendo una profunda transformación social gracias a las tecnologías de la información. Como se indica en “Evolución tecnológica e impacto laboral y socio-económico de los videojuegos” [1] esto se debe a la consolidación de la televisión e internet así como a las aplicaciones móviles y por supuesto a los videojuegos. Además los videojuegos se han consolidado como una innovación de referencia que, además de convertirse en un motor económico, está modificando la adquisición y transmisión de conocimiento.

Como indican en “Introducción a los videojuegos” y “Breve historia de los videojuegos” [2 y 3] el inicio de los videojuegos es difuso y además esto se debe al propio concepto de videojuego. Sin embargo es posible fijar su comienzo en torno a la década de los cincuenta del siglo XX con videojuegos como "Cathode-Ray Tube Amusement Device" desarrollado en 1948 por Thomas T. Goldsmith, Cedar Grove y Estle Ray o como “OXO”, desarrollado por Alexander S. Douglas en 1952. A partir de estas fechas se han creado videojuegos de multitud de géneros diferentes (aventuras, plataformas, multijugador, carreras, etc) y para multitud de dispositivos, incluyendo en la actualidad dispositivos móviles.

Como se dice en “Hábitos de uso de los videojuegos en España entre los mayores de 35 años” [4] los videojuegos demuestran un creciente grado de implantación social, particularmente en los países más desarrollados debido en parte a los dispositivos móviles. Además tienen un impacto en la totalidad de la sociedad y no solo en los grupos más jóvenes.

Las aplicaciones móviles están muy presentes en la sociedad y como se marca en “Tendencias evolutivas del contenido digital en aplicaciones móviles” [5] los videojuegos son una tendencia de las mismas por encima en ocasiones de las redes sociales. Con este proyecto se pretende contribuir a que esta situación se mantenga o siga aumentando.

Figura 1: Videojuegos de plataformas/aventuras

1.1.2 Justificación

El objetivo principal del proyecto es el desarrollo de un videojuego de plataformas 2D desde cero. Actualmente el mundo del ocio en dispositivos móviles es extenso pero en este tipo de videojuegos la novedad es importante. Los videojuegos de plataformas suelen ser juegos que se descargan, se juegan y se desinstalan constantemente.

Umbrellibur se presenta como un videojuego ambientado en la época medieval donde se sustituye la historia de la espada “Excalibur” por la historia del paraguas “Umbrellibur”. Consiste en un conjunto de rompecabezas y enemigos dentro de un ambiente de videojuego 2D de plataformas. Con este paraguas se pueden realizar ciertos movimientos como atacar, volar, protegerse, apagar fuegos utilizando el agua...

Lo que marca la diferencia en *Umbrellibur* es la creación de una nueva mecánica de juego basada en un objeto cotidiano y que se usa de manera intuitiva.

Se plantea un desarrollo desde cero en el que se seguirán las fases citadas en “Metodología para el desarrollo de aplicaciones móviles” [6] que son análisis, diseño, desarrollo, pruebas de funcionamiento y entrega. Además, como se indica en “Usabilidad en aplicaciones móviles” [7] la usabilidad debe ser un factor fundamental para garantizar un producto software de calidad.

1.1.3 Estudio de mercado

En este apartado se van a analizar las alternativas similares al videojuego *Umbrellibur* que se encuentran en el mercado.

Los videojuegos pertenecen a un sector en constante evolución. A lo largo de su historia los videojuegos de plataformas han tenido un peso bastante relevante para quedar actualmente en un plano más secundario. Actualmente los juegos multijugador online ocupan un puesto relevante en el sector.

Aun así, muchos juegos de plataformas se encuentran gran en el playStore. Prácticamente salen juegos nuevos cada semana y cada uno con sus particularidades.

Se pueden citar ciertos juegos de plataformas actuales para dispositivos móviles de los que se pueden tomar ideas, tanto estéticas como funcionales: *Typoman Mobile*, *Rayman Adventures* o *Red ball* son ejemplos de ello.
También la estética stickman está muy presente en este sector con juegos como *Stickman parkour platform* o *Stickman Soccer*.
Además se han encontrado juegos que utilizan un paraguas de forma similar como *Umbrella Master* o *Umbrella Falling* en los que se utiliza el paraguas para volar exclusivamente y no para golpear o apagar fuegos.

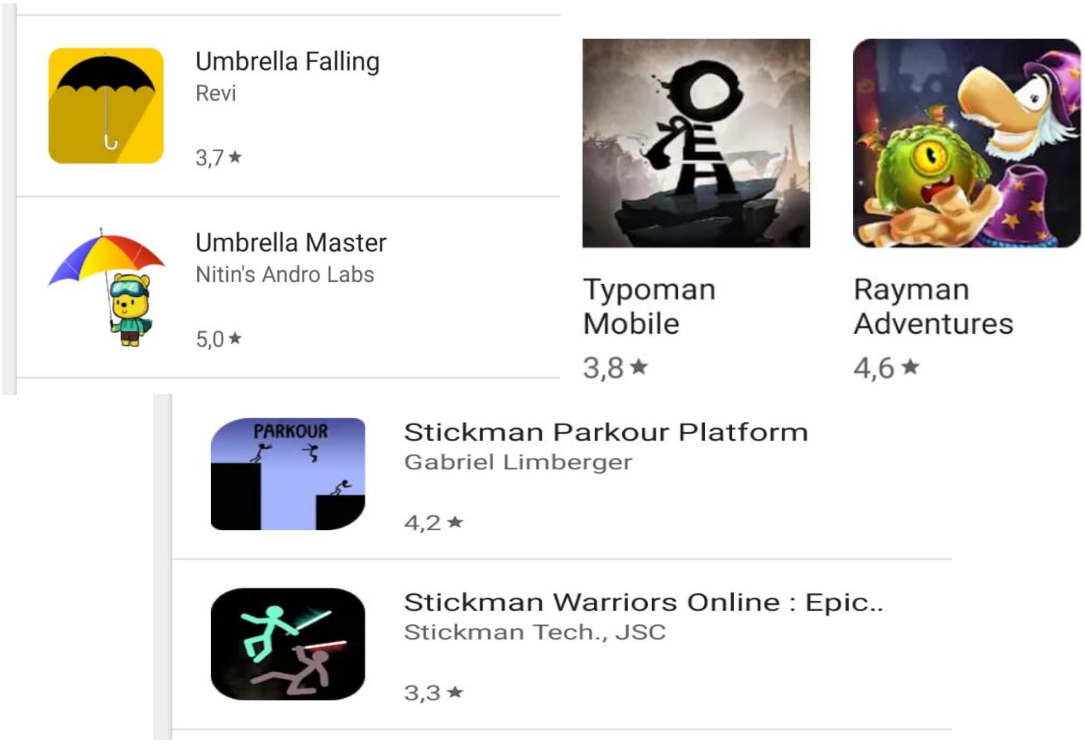


Figura 2: Videojuegos similares

1.2 Objetivos del trabajo

El objetivo principal es desarrollar (analizar, diseñar e implementar) un videojuego de plataformas 2D para dispositivos móviles que sea original y divertido.

Además de este objetivo se debe:

- Poner en práctica los conocimientos adquiridos a lo largo de todo el máster.
- Definir una planificación realista para un proyecto, teniendo en cuenta su alcance y los recursos disponibles.
- Ser capaz de documentar y justificar las decisiones tomadas en el desarrollo y los resultados conseguidos.
- Adquirir experiencia en afrontar los retos que supone sacar adelante un proyecto completo.

1.3 Enfoque y método seguido

1.3.1 Tecnologías utilizadas

El producto de este TFM será un videojuego para dispositivos móviles. Para el desarrollo del mismo son necesarias ciertas tecnologías que se detallan a continuación:

- Programación: **Unity**. El videojuego se desarrolla utilizando Unity 2D debido a la facilidad y gracias al conocimiento adquirido en la asignatura "introducción a videojuegos en dispositivos móviles". Se elige este motor en lugar de Unreal Engine o en lugar de la programación nativa (Android o IOS) debido a la facilidad de exportar el proyecto a todo tipo de plataformas y debido a que ya se ha realizado una introducción a la tecnología. Pese a esta introducción será necesario ampliar el conocimiento en Unity.
- Entorno de desarrollo: **Visual Studio**. Este es el IDE utilizado en el proyecto debido a que es el más recomendado para desarrollar en Unity.
- Gráficos: **Inkscape**, **Pixlr** y **Gimp**. Se utilizan estas herramientas de software libre que permitirán la creación de gráficos necesarios en el videojuego debido a que son suficientes para realizar la mayor parte de las acciones necesarias. Para utilizarlas se ha requerido un tiempo de formación.
- Documentación: **MS Office**. Se utiliza el paquete Office para realizar documentos como la memoria.

Figura 3: Logos de las herramientas utilizadas

1.3.2 Diseño

La estética del juego será la siguiente:
Dadas las limitaciones artísticas del autor se decide diseñar el personaje como un “Stickman” con un paraguas.
El resto del decorado se realizará utilizando imágenes reales y aplicando ciertos filtros para dejarlos en escala de grises. De esta forma se puede conseguir una estética siniestra y bastante identificable (ver figura). Los enemigos humanos seguirán la estética “Stickman” mientras que si se utilizan animales se utilizarán imágenes reales.

Figura 4: Captura de Umbrellibur

1.3.3 Metodología

La metodología que suele utilizarse en proyectos de software con un marcado inicio y fin suele ser la metodología en cascada. Al ser un desarrollo de un proyecto desde cero y con grandes posibilidades de ampliación es muy recomendable utilizar metodologías ágiles para poder ir modificando en ciertos "sprints" los diferentes niveles, la historia, los movimientos... con el objetivo de conseguir con un juego entretenido y atractivo.

Estas metodologías se utilizarán de manera combinada utilizando como "sprints" las diferentes entregas.

Cascada:
El modelo en cascada es un proceso de desarrollo secuencial, en el que el desarrollo de software se concibe como un conjunto de etapas que se ejecutan una tras otra.

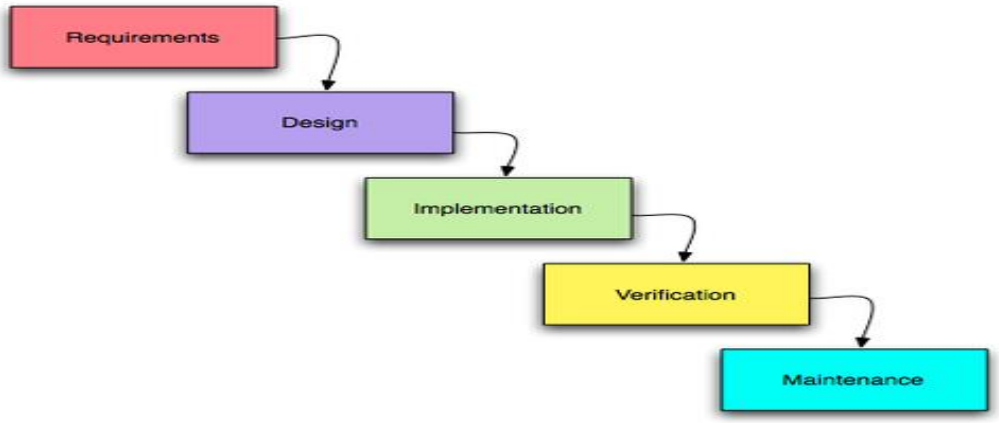


Figura 5: Cascada
*Fuente: https://cdn-images-1.medium.com/max/800/0*IzcrwKc3tXqje4r-.png*

Scrum:
En scrum se realizan una serie de iteraciones o sprints a través de las cuales se va perfilando el producto final (ver figura).

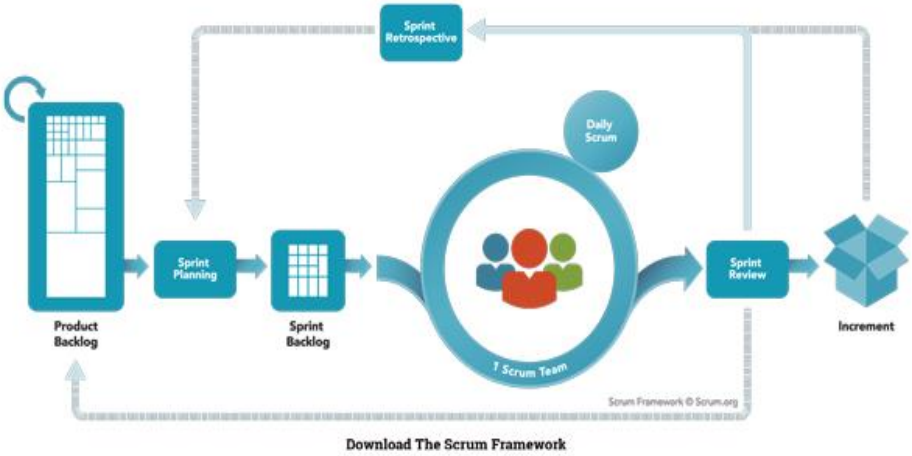


Figura 6: Scrum
Fuente: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRLjgmN2bq_wW3Udmj94W9CYCBede50w6ZLSmCtfyREaKNzHWPs

1.4 Planificación del trabajo

1.4.1 Recursos

PC:

- Acer Aspire E1-571
- Macbook Air pro
- Xiaomi MiA1

Programas:

- Unity
- Pixlr
- Inkscape
- Visual Studio
- MS Office
- Gantt proyect
- Wireframe
- Firebase

Personal:

- Pablo Pérez Martínez, diseñador y desarrollador.
- Francesc d'Assís Giralt Queralt, profesor consultor.

1.4.2 Planificación

El TFM está dividido en 4 entregas:

- PEC1: **13/03/19** Planificación
- PEC2: **03/04/19** Diseño
- PEC3: **15/05/19** Implementación
- PEC4: **05/06/19** Entrega final

Siguiendo estas entregas se puede agrupar el trabajo en 4 grandes bloques:

- Planificación: se realizará una planificación con estimación de tiempos y costes y se describirá la metodología de trabajo.
- Diseño: se realizará un diseño centrado en el usuario.
- Implementación: se implementará la aplicación y se pasarán pruebas.
- Entrega: se realizará un entregable final y se subirá al Market.

En la siguiente figura se puede observar el desglose de tareas junto a su "coste" estimado en horas.

Nom	Date de début	Date de fin	Cost
Planificación	20/02/19	13/03/19	
Selección de tema	20/02/19	21/02/19	5
Definición de objetivos	21/02/19	07/03/19	10
Definición del enfoque y ...	21/02/19	07/03/19	6
Diagramas de planificaci...	07/03/19	07/03/19	4
Documento PEC1	20/02/19	13/03/19	20
Diseño	01/03/19	03/04/19	
Análisis de usuarios	07/03/19	12/03/19	20
Diseño de elementos	01/03/19	28/03/19	15
Preparación de recursos	01/03/19	28/03/19	50
Prototipo en papel	18/03/19	25/03/19	15
Prototipo funcional	07/03/19	03/04/19	30
Documento PEC2	13/03/19	03/04/19	20
Implementación	25/03/19	15/05/19	
Interfaz gráfica	01/04/19	15/04/19	15
Funcionalidad del perso...	25/03/19	15/04/19	45
Niveles	15/04/19	25/04/19	45
Historia	01/04/19	15/04/19	15
Pruebas y depuración	22/04/19	07/05/19	20
Documento PEC3	03/04/19	15/05/19	20
Entrega	15/05/19	05/06/19	
Redacción y revisión de l...	15/05/19	02/06/19	25
Presentación	20/05/19	02/06/19	10
Vídeos	20/05/19	02/06/19	10
Entrega final	15/05/19	05/06/19	20

Figura 7: Planificación

En la siguiente figura se observa el diagrama de Gantt con la planificación del proyecto.

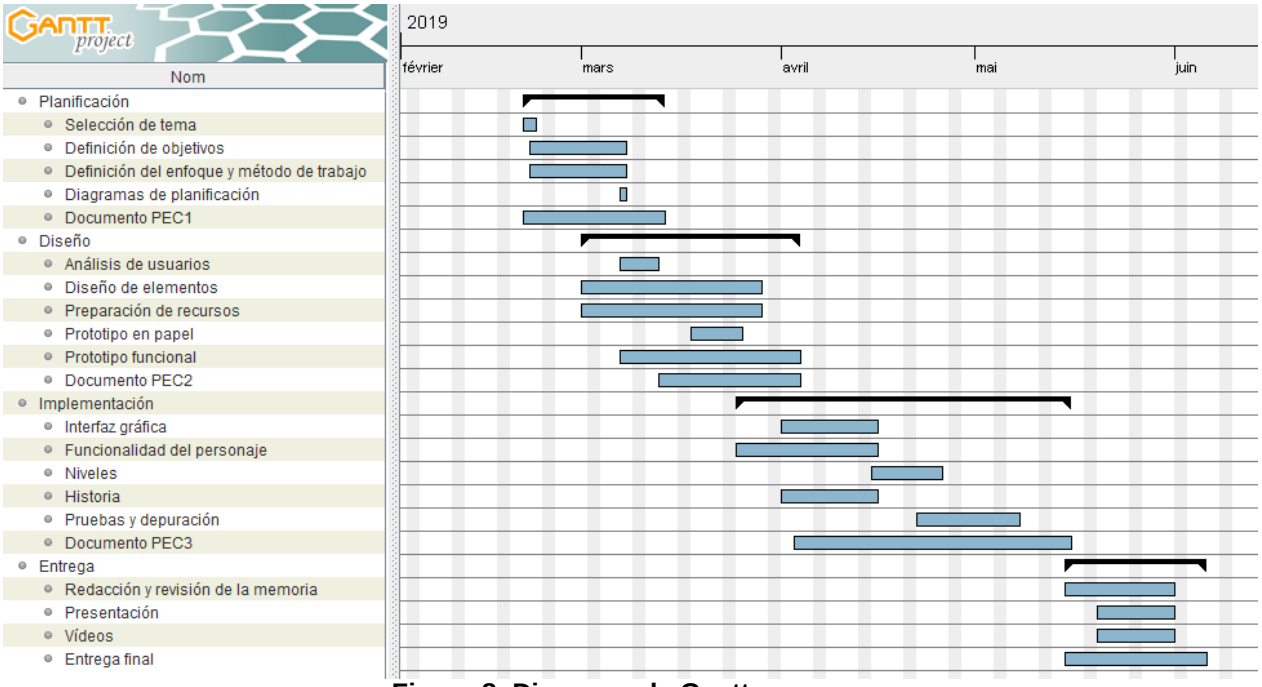


Figura 8: Diagrama de Gantt

El proyecto se desarrolla desde el 20 de febrero hasta el 5 de junio de 2019 con una duración total de 105 días. Siguiendo esta planificación se requerirán 420 horas aproximadamente, lo que, dividido entre el número de días, es una media de 4 horas al día.

La dedicación estimada será la siguiente: 5 horas al día en días laborables (o festivos entre semana) y 7 los fines de semana. (Se tendrá en cuenta que habrá días que por motivos personales o laborales no se podrá dedicar ninguna hora al proyecto).

1.4.3 Estimación de costes

Equipo informático:

Acer Aspire E1-571 (0€)

Intel core i7-3612QM 2.1GHz	Intel HD Graphics 4000
15,6” HD LED LCD	DVD-Super Multi DL Drive
8 GB DDR3	Acer Nplify 802.11b/g/n
750 GB HDD	6 cell Li-ion battery

MacBook Air (0€)

Intel core i5 1.3GHz	Intel HD Graphics 5000
11”	128GB SSD
4 GB RAM	

Cuenta de Play Store: 25€
Software: Software Libre 0€
Horas de trabajo: 420€ x 15€/h = 6300€
Se estimaría en un total de 6325€, gracias a contar con el equipo informático.

1.4.4 Análisis de riesgos

En este apartado se analizarán los riesgos que se deben tener en cuenta para que se realice correctamente el trabajo y se cumpla con la planificación.

Riesgo 1: Poco tiempo y entregas fijas
Descripción: Debido a la naturaleza del trabajo de fin de máster se tiene unos plazos limitados para realizar las entregas.
Plan de acción: Se debe planificar correctamente cada una de las entregas como medida preventiva, y, en caso de no llegar a cumplir los hitos, priorizar ciertas tareas llegando al objetivo a tiempo (sin excluir detalles importantes).

Riesgo 2: Falta de conocimiento de la tecnología
Descripción: El proyecto parte con la base fundamental de la asignatura Introducción a videojuegos en dispositivos móviles. En esta asignatura no ha sido necesario centrarse en el dispositivo sino en los videojuegos y ciertas opciones han quedado fuera de la materia por falta de tiempo.
Plan de acción: Se deberá dedicar cierto tiempo a la formación así como intentar ajustar los requisitos y la funcionalidad de la aplicación al conocimiento que ya se posee de la tecnología.

Riesgo 3: Falta de recursos (gráficos, sonoros…)
Descripción: El proyecto parte desde cero por lo que es necesario una gran cantidad de recursos gráficos, sonoros o fuentes de texto.
Plan de acción: Se debe planificar el proyecto con una cierta cantidad de horas dedicadas a la producción de estos recursos. Estas horas pueden aumentar el tiempo por encima de lo habitual en un TFM promedio pero el resultado será más personalizado que utilizando recursos de terceros. Si el tiempo es demasiado grande se podrá recurrir en ciertos casos a recursos de terceros.

Riesgo 4: Bugs o problemas de programación complejos
Descripción: Siempre que se programa suelen aparecer “Bugs”, errores de programación debidos a diversos factores como el desconocimiento de la tecnología, el desarrollo excesivamente rápido, etc.
Plan de acción: Se deben tener en cuenta ciertas horas para solucionar estos problemas en la planificación. Si resolverlos requiere demasiado tiempo es posible dar una vuelta al problema e intentar resolverlo de otra forma.

Riesgo 5: Cambios en la situación personal
Descripción: La situación personal puede ser un factor clave a la hora de desarrollar el TFM. Todo puede afectar a su correcto desarrollo, un cambio en una relación familiar o sentimental, un viaje no planificado, un cambio laboral…
Plan de acción: Se deben tener en cuenta ciertas horas fijas para el desarrollo del trabajo e intentar que este tipo de riesgos no afecten negativamente al desarrollo. Dependiendo de la gravedad podría ser desde un riesgo leve hasta un riesgo muy grave.

Los riesgos detallados anteriormente se podrían clasificar en función de su probabilidad e impacto de la siguiente forma:

Impacto/ Probabilidad	Muy poco probable	Poco probable	Probable	Muy probable
Bajo			R1, R2	
Moderado				R3
Muy grave	R5		R4	

1.5 Breve sumario de productos obtenidos

Los productos que se habrán obtenido al finalizar del trabajo son los siguientes:

- Memoria
- Código fuente
- Recursos (sprites, audios…)
- APK
- Vídeo para presentar el proyecto
- Vídeo para demostrar el funcionamiento de la App
- Diapositivas para la presentación del TFM

1.6 Breve descripción de los siguientes capítulos de la memoria

La estructura de la memoria sigue el siguiente esquema:

Introducción:
En este primer capítulo se ha abordado la primera fase del proyecto que consiste en la elección del proyecto y la planificación del diseño e implementación de la aplicación.

Análisis:
En el segundo capítulo se investigarán los usuarios de la aplicación y se examinarán las condiciones en las que será utilizada la aplicación.

Diseño:
En el tercer capítulo se generará un primer prototipo en papel y un primer prototipo del videojuego (funcional), destacando los detalles de diseño de la aplicación.

Implementación
Contiene la descripción del proceso de implementación: la selección de herramientas, resolución de problemas, diagramas...

Pruebas:
Una vez terminada la implementación y antes de realizar la subida al Market se definirán una serie de pruebas para verificar el correcto funcionamiento de la aplicación.

Publicación:
Contiene el proceso de subida de la aplicación a Google Play Store.

Conclusiones:
Este apartado recoge las conclusiones sobre el TFM, una reflexión crítica sobre el logro de los objetivos planteados inicialmente y un análisis crítico del seguimiento de la planificación y metodología a lo largo del producto. Además se marcarán las futuras líneas de desarrollo para la aplicación.

2. Análisis

El objetivo del trabajo es realizar un diseño centrado en el usuario. Por ello, es necesario realizar un análisis de los usuarios potenciales y de la funcionalidad requerida antes de realizar el diseño.

2.1 Análisis de usuarios

Se parte de la idea de que los principales usuarios potenciales del videojuego serán personas menores de 40 años pero, para acotar más a los usuarios potenciales del videojuego, se ha realizado una encuesta en la que se analiza el tipo de perfil del usuario y lo que puede buscar en este tipo de aplicaciones.

Se ha propuesto un formulario a través de la herramienta Google Forms que permite de forma gratuita realizar encuestas.
Se ha difundido el enlace en internet de manera que se consiguen unos datos de personas que lo realizan sin conocer el objetivo del estudio ni el estilo del videojuego pero que, por otro lado, puede ser poco fiable debido a que no se filtra ni contrasta cómo se ha rellenado.
Este formulario se encuentra en el apartado de anexos.

A continuación se muestran los resultados de la encuesta:

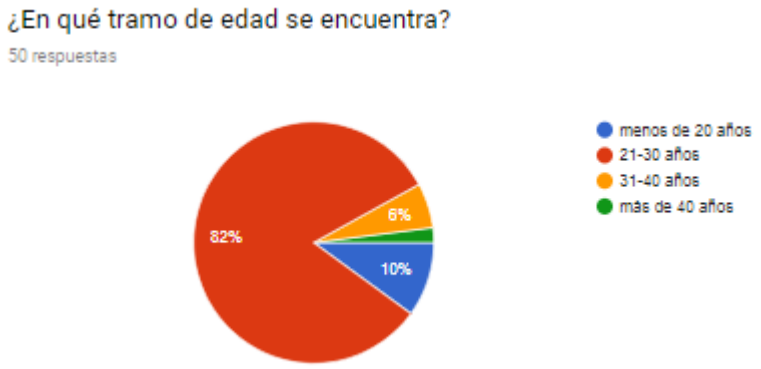


Figura 9: Tramo de edad

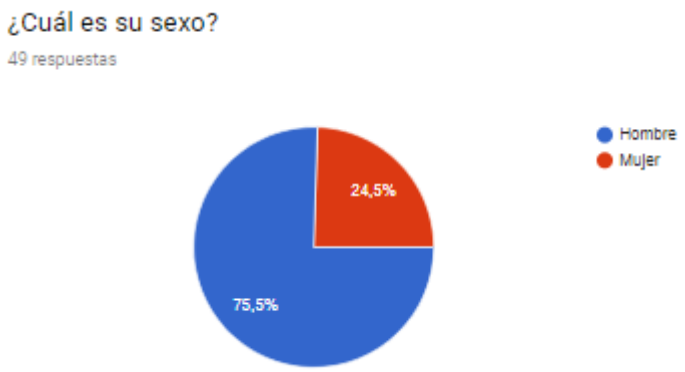


Figura 10: Sexo

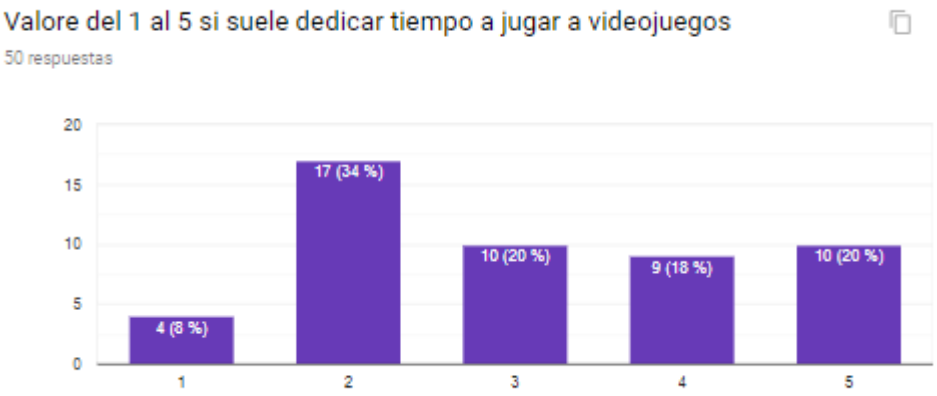


Figura 11: Tiempo dedicado a videojuegos

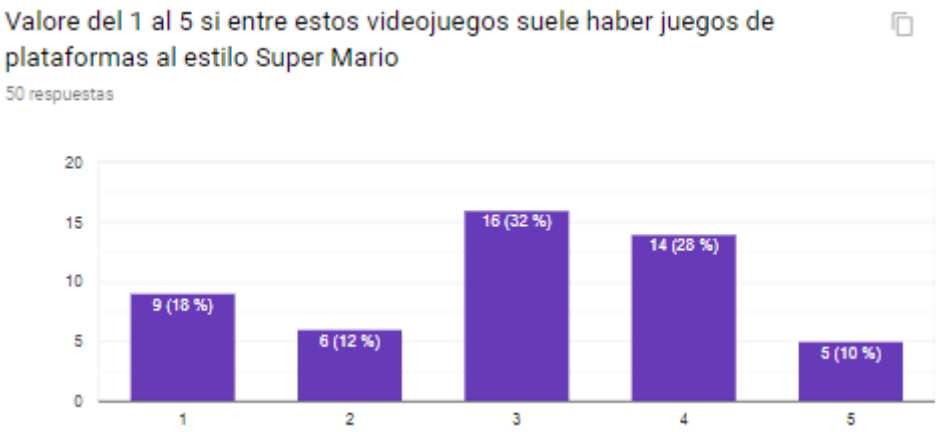


Figura 12: Dedicación a juegos de plataformas

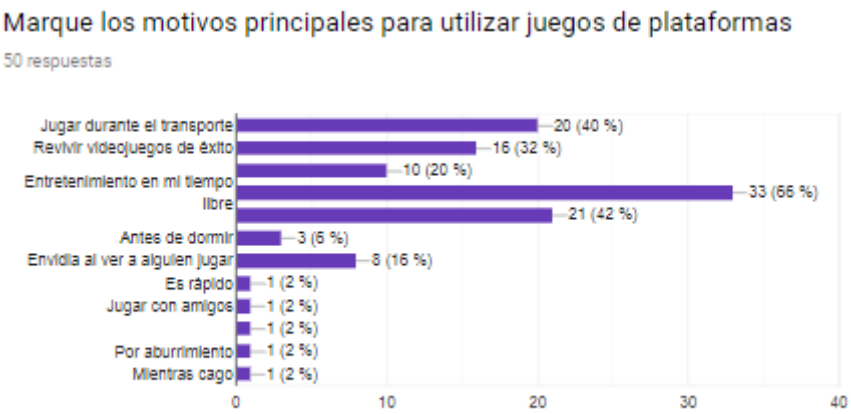


Figura 13: Motivos de utilización

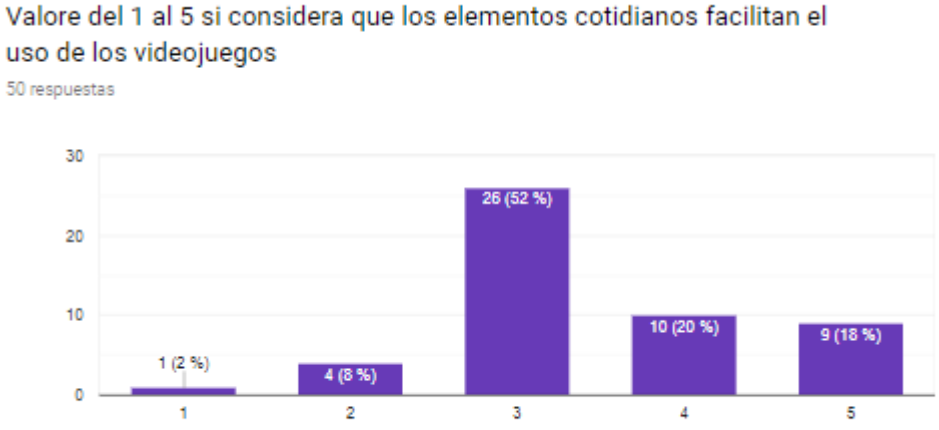
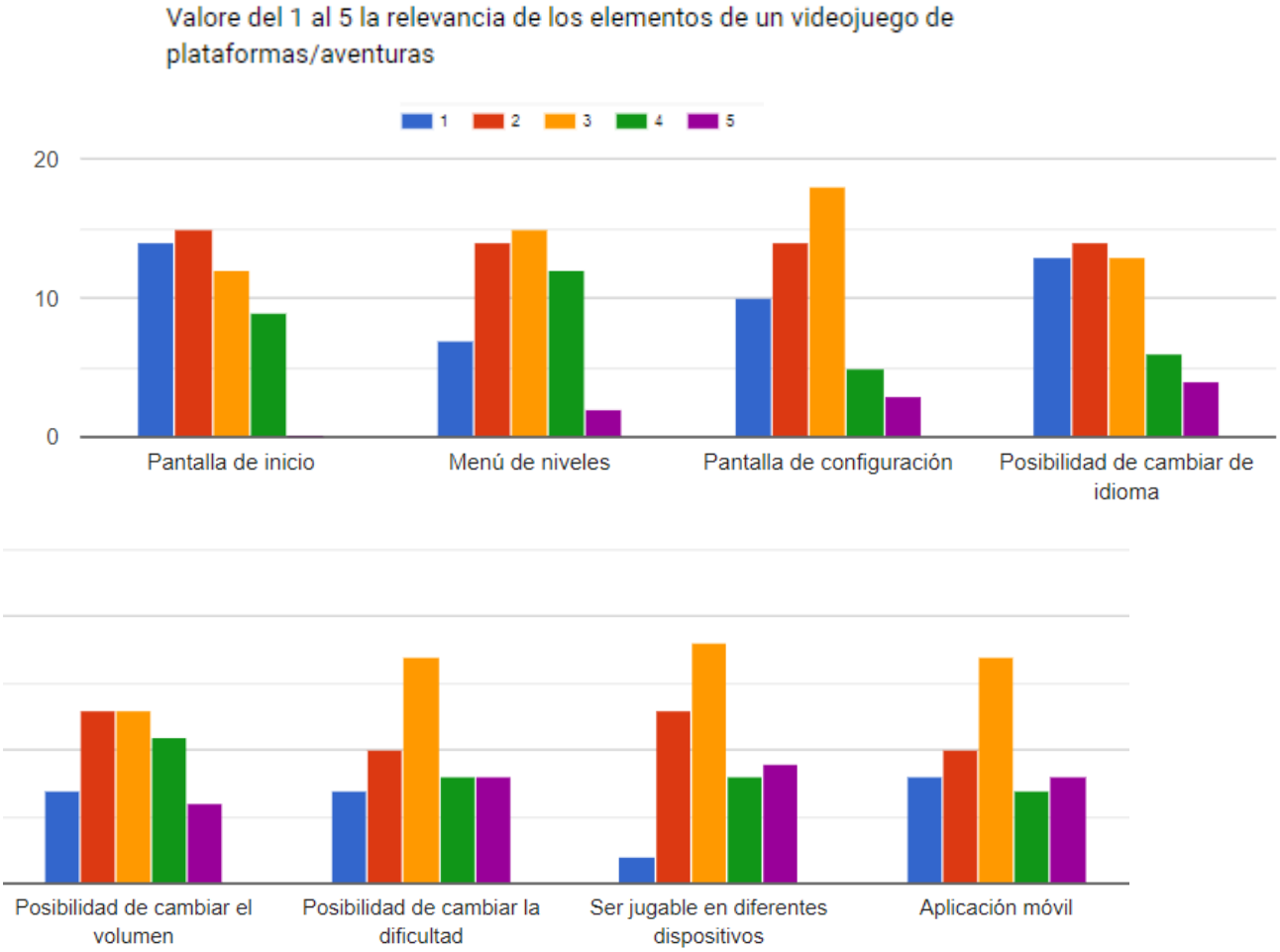


Figura 14: Elementos cotidianos en videojuegos



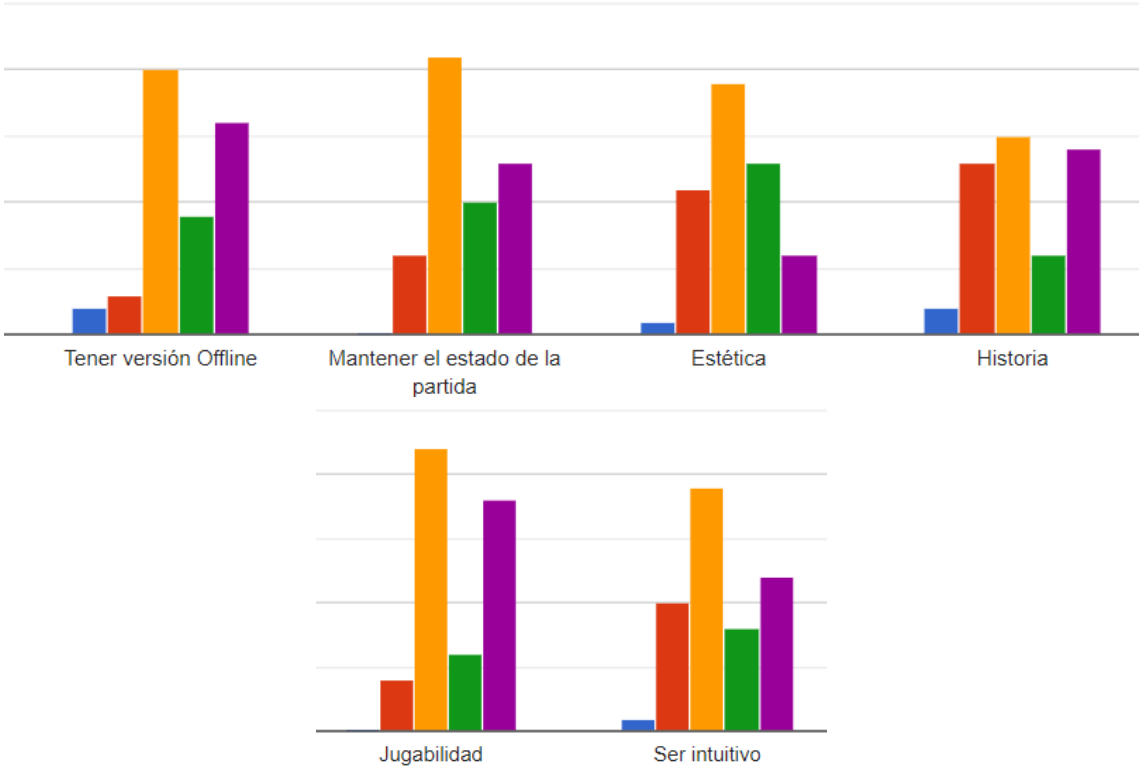


Figura 15: Funcionalidad deseada

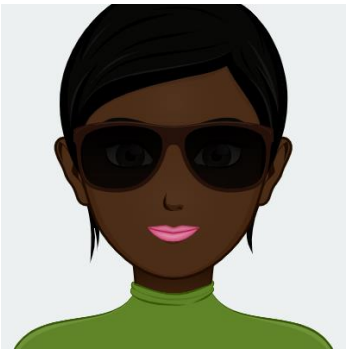
Loe resultados obtenidos se han analizado y se han extraído las siguientes conclusiones:
De los 50 encuestados la mayor parte 29 considera que dedica cierto tiempo a los videojuegos y solo 4 considera que no dedica tiempo. De entre todos los encuestados el 38% suele incluir a menudo en su tiempo libre juegos de plataformas, aun así un 18% (9) no juega casi nunca a este tipo de juegos.
Los jugadores que juegan poco a videojuegos son los que más dedican su tiempo a juegos de plataformas y los que más juegan “por envidia”.
Los principales motivos o las principales situaciones para jugar a videojuegos de plataformas son: durante el tiempo libre (33 de 50) o para desconectar (21) y durante el transporte (20). Un 32% los utiliza también para evocar recuerdos (revivir videojuegos de éxito). Además hay otras situaciones como jugar al ver a otro usuario jugar (16%) o para inspirarse artísticamente (20%).
Solo un 10% de los encuestados considera que los elementos cotidianos no facilitan el uso de videojuegos, mientras que el 38% (19) considera lo contrario.
En cuanto a los elementos más necesarios para un correcto videojuego de plataformas lo más demandado es una buena jugabilidad y una versión offline unido a una buena historia y a ser intuitivo. En cuanto a las diferentes pantallas (configuración, inicio, niveles…) no existe ninguna que sea imprescindible. La estética y el correcto guardado del estado de la partida también son elementos importantes para la mayor parte de los encuestados.
Además de estos datos cuantitativos se han realizado ciertas entrevistas informales y desestructuradas para profundizar en el análisis de los usuarios y los requisitos deseados.

2.2 Fichas de usuario y escenarios de uso

En este apartado se presentan usuarios potenciales asociados a un escenario de uso posible extraídos a partir de la encuesta:

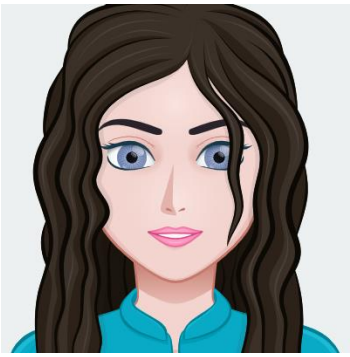
Nombre: Mario
Edad: 28
Ocupación: Desarrollador de videojuegos
Descripción:
Apasionado de los videojuegos desde niño. Actualmente se dedica a desarrollar videojuegos para una gran empresa del sector.

Escenario de uso:
Mario se dedica a desarrollar videojuegos. Decidió estudiar para poder desarrollar videojuegos porque era su pasión desde pequeño. Sin embargo actualmente no tiene demasiado tiempo para jugarlos. Como tiene poco tiempo suele buscar de vez en cuando un juego de plataformas con una historia corta y una funcionalidad atractiva para inspirarse y desconectar al mismo tiempo.



Nombre: Dafne
Edad: 32
Ocupación: Ingeniero industrial
Descripción:
Poco tiempo libre.
Trabaja en una gran consultora a la que va en transporte público.

Escenario de uso:
Dafne tiene mucho estrés, trabaja en una gran consultora y además suele tener que ayudar a su hermana cuidando a su cachorro. Le encantaban los videojuegos cuando era niña y actualmente suele utilizar el transporte a su trabajo para jugar a videojuegos sencillos y que no le exijan mentalmente.



Nombre: Cristina
Edad: 23
Ocupación: Profesora de educación primaria
Descripción:
Trabaja con niños.
Acaba de terminar sus estudios.

Escenario de uso:
Cristina suele llegar a casa cansada y suele tener que preparar clases para sus alumnos. Como los alumnos están muy inmersos en las nuevas tecnologías juega a ciertos videojuegos en su tiempo libre para mantenerse al día y (mientras se divierte) preparar premios para sus clases relacionados con esos videojuegos.

Nombre: Pablo
Edad: 25
Ocupación: Trabajador social
Descripción:
Es una persona poco interesada en los juegos. Le interesan las relaciones personales más que los dispositivos móviles.

Escenario de uso:
Pablo es una persona poco interesada por los móviles. Compró su primer Smartphone de segunda mano mucho después que sus amigos. De vez en cuando ve a alguno de sus amigos jugando y le da envidia y prueba los juegos. No suele ser habitual pero en ocasiones se los descarga.

2.3 Requisitos

Una vez analizados los usuarios es conveniente fijar los requisitos funcionales y no funcionales más importantes para desarrollar un videojuego correcto en este Trabajo de Fin de Máster.

Los requisitos se separan en tres bloques por orden de prioridad:

- Mínimo aceptable para el TFM (M)
- Deseado para un buen TFM (D)
- Funcionalidades opcionales (o de desarrollo futuro) (O)

Los requisitos funcionales de la aplicación son:

- Interfaz de usuario (M)
- Pantalla de niveles (M)
- Movimiento del personaje (M)
- Botones (M)
- Programación de movimiento (M)
- Programación de salto (M)
- Programación de ataque (M)
- Programación de funcionalidad del paraguas básica (M)
- Pantalla de configuración (D)
- Programación de funcionalidad del paraguas avanzada (D)
- Múltiples niveles (D)
- Guardado del niveles superados (D)
- Integración de Firebase con notificaciones (O)
- Subida al Market (O)
- Integración de Firebase con publicidad (O)
- Integración de Firebase con perfil de usuario (O)
- Logros (O)

Los requisitos no funcionales son:

- Versión offline
- Disponibilidad multiplataforma
- Interfaz táctil
- Interfaz sencilla e intuitiva
- Facilidad de mantenimiento y evolución del software
- Diseño atractivo
- Estética atractiva
- Historia entretenida

2.4 Casos de uso

En este apartado se muestran los diferentes casos de uso que se extraen del análisis de los usuarios y de los requisitos de la aplicación así como de los diagramas de casos de uso.

2.4.1 Descripción de los casos de uso

Caso de uso 1	**Instrucciones**
Descripción	El usuario quiere conocer en detalle el funcionamiento del juego por lo que entra en la pantalla de instrucciones para aprender
Precondiciones	Usuario en la pantalla de menú.
Situación final	El usuario resuelve sus dudas y vuelve al menú.

Caso de uso 2	**Opciones**
Descripción	El usuario desea cambiar la configuración del juego
Precondiciones	Configuración inicial (primera vez por defecto)
Situación final	Configuración elegida por el usuario

Caso de uso 3	**Jugar un nivel**
Descripción	El usuario quiere jugar un nivel para continuar con la historia y completar el juego
Precondiciones	Usuario en el menú
Situación final	Menú de niveles con un nivel más desbloqueado, varios o ninguno dependiendo de los niveles superados.

Caso de uso 3.1	**Movimiento (Dentro del juego)**
Descripción	El usuario quiere mover al personaje principal
Precondiciones	Usuario en un nivel
Situación final	El personaje realiza la acción que ha pulsado en los botones (izquierda, derecha, salto, abrir, escudo o ataque)

2.4.2 Diagramas de casos de uso

A continuación se muestran los diagramas de los casos de uso de la aplicación explicados anteriormente.

En el primer diagrama se muestra un actor o usuario que tiene 4 opciones (jugar, configurar la App, leer las instrucciones o salir).

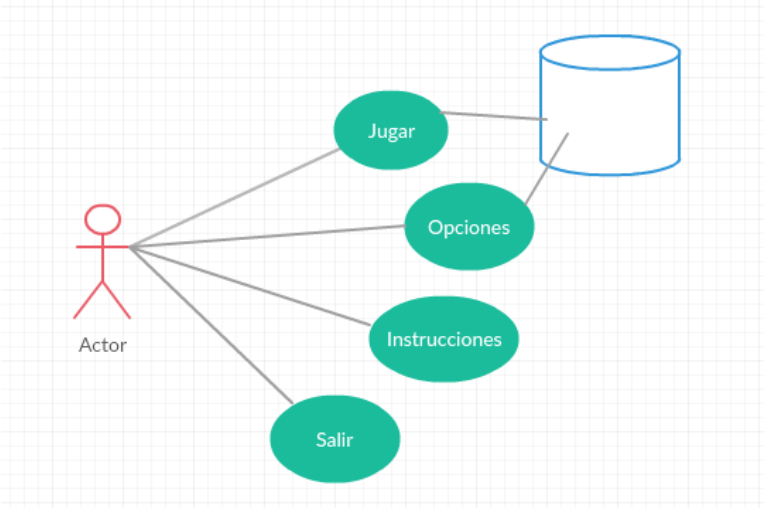


Figura 16: Casos de uso

En el segundo diagrama se detallan las posibilidades del usuario en el caso de uso de jugar. Podrá seleccionar un nivel y dentro del nivel podrá utilizar las diferentes opciones para pasarlo.

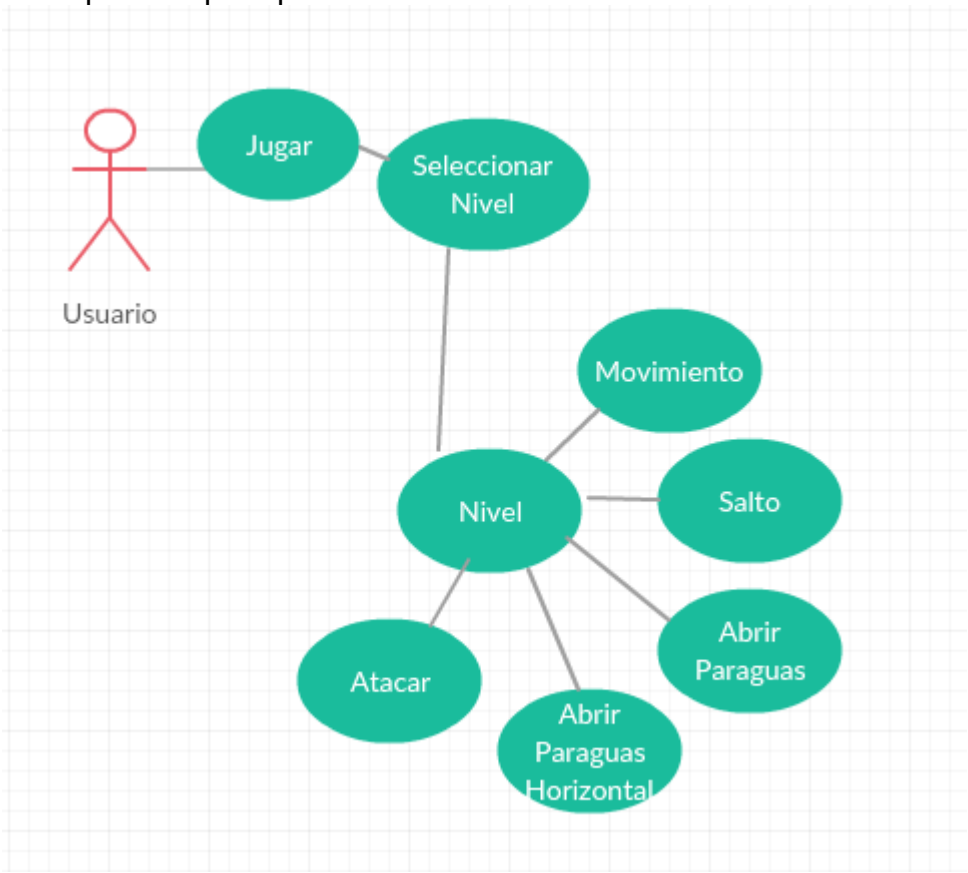


Figura 17: Caso de uso: Jugar

3. Diseño

Una vez desarrollado el análisis se procede a diseñar la aplicación. En este capítulo se mostrarán diferentes apartados del diseño del videojuego: la arquitectura, el mapa de navegación, el diseño estético y el prototipo realizado.

3.1 Arquitectura

Según se ha mencionado en el capítulo de introducción, el videojuego se desarrolla utilizando Unity que permite un desarrollo cross-platform y facilitará la migración a cualquier tipo de dispositivo.

La arquitectura del proyecto se plantea de la siguiente forma:

Se agrupan ciertas escenas que formarán la interfaz de usuario (UI): el menú, la pantalla de configuración, el menú de niveles y las instrucciones.
Además existirán escenas diferentes por cada nivel, en las que se irán situando ciertos Prefabs con los elementos (personaje, trampas, lluvia, etc.)

Los niveles del juego se presentan con un personaje principal que usará un patrón “singleton” para facilitar su acceso desde otros elementos y una cámara que llevará asociados los botones y seguirá al personaje principal.

Es necesario un sistema de persistencia de datos. Dado que el nivel de datos que hay que guardar en la aplicación es reducido, se utiliza el sistema proporcionado por Unity: “PlayerPrefs”.

3.2 Mapa de navegación

El mapa de navegación de la aplicación se muestra a continuación: un menú con cuatro opciones (opciones o configuración, instrucciones, jugar y salir) y una vez situados en el menú de niveles la opción de elegir un nivel. Desde un nivel x se podrá pasar a un nivel x+1 o bien volver al menú de niveles. Con este mapa de navegación pueden realizarse los casos de uso del programa.

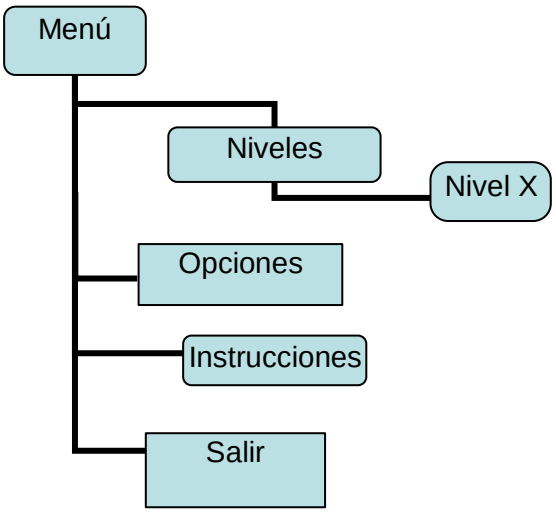


Figura 18: Mapa de navegación

3.3 Prototipo de baja fidelidad

En este apartado se muestra un primer prototipo de baja finalidad que se ha trazado utilizando la herramienta online https://wireframe.cc/.
En este prototipo se observan 4 pantallas que cumplen con el mapa de navegación: el menú principal, el menú de selección de nivel, el menú de configuración y la pantalla de juego (en la que se encuentran los botones que puede utilizar el usuario).

Menú principal: siguiendo el mapa de navegación es la pantalla que abrirá la App y que permitirá entrar en el menú de niveles, en las opciones, en las instrucciones y salir de la misma.

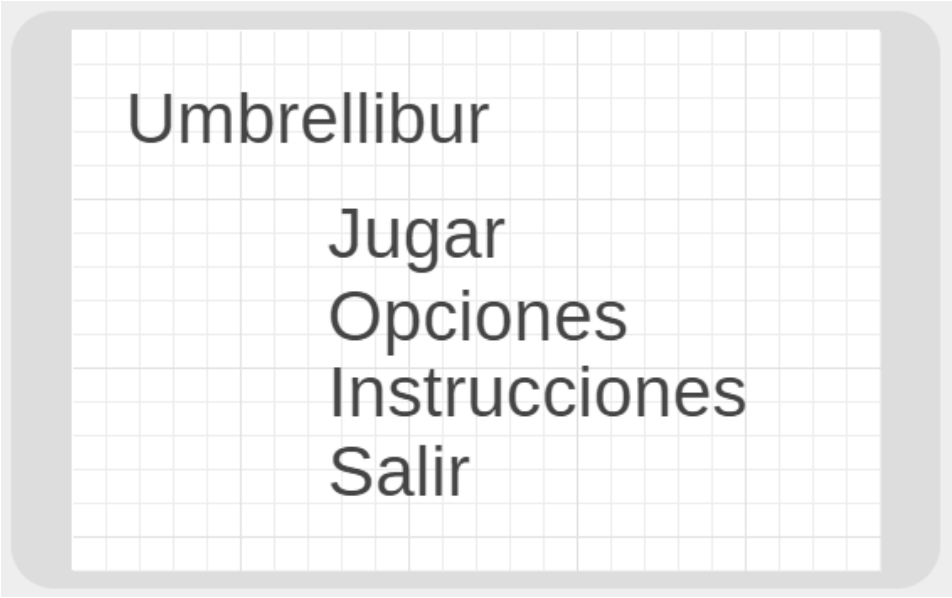


Figura 19: Prototipo baja fidelidad pantalla inicial

Selección de nivel: permitirá abrir los niveles que se hayan desbloqueado.

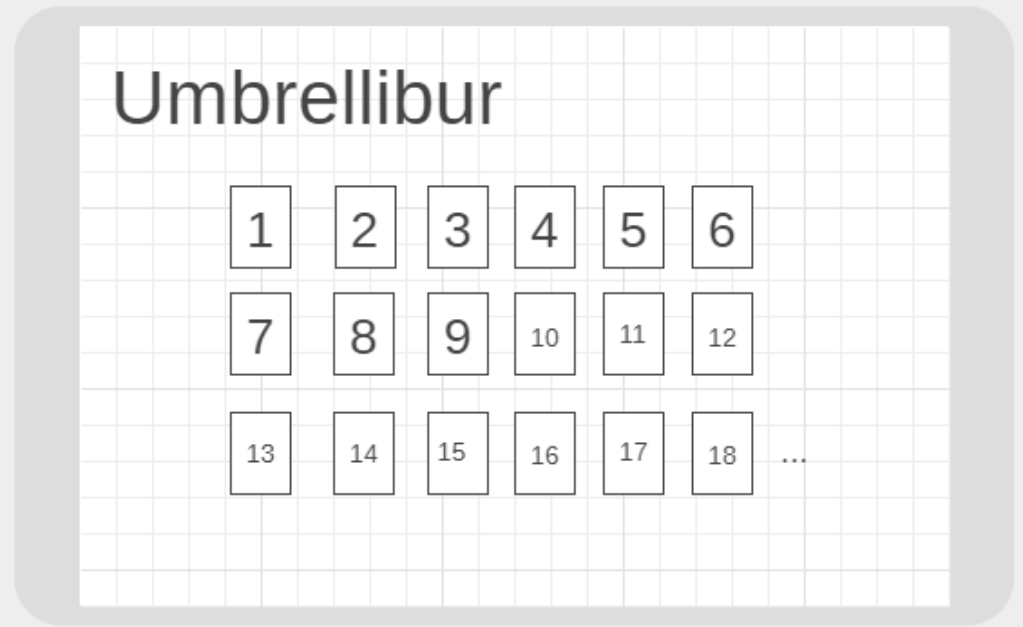


Figura 20: Prototipo baja fidelidad selección de nivel

Opciones: permitirá cambiar ciertos parámetros de la aplicación.

Figura 21: Prototipo baja fidelidad opciones

Nivel: se permitirá controlar al personaje mediante botones.

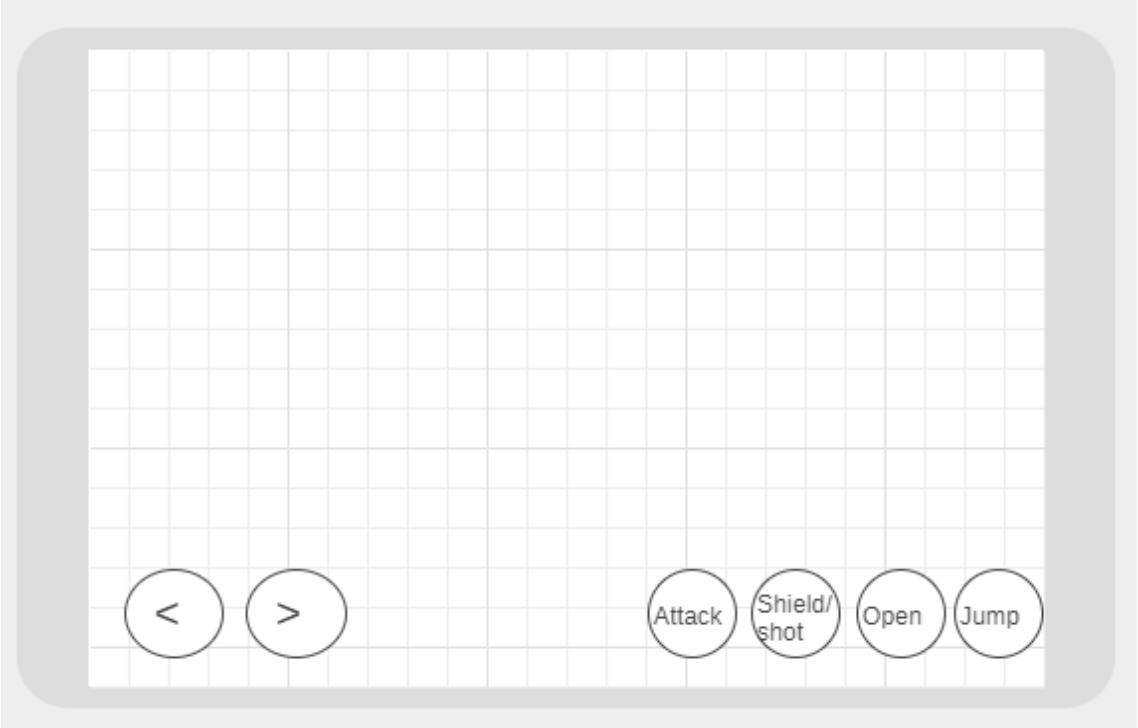


Figura 22: Prototipo baja fidelidad nivel

3.4 Diseño de elementos

La estética de un videojuego es un componente esencial del mismo. Es por esto que es necesario preparar unos recursos adecuados. Como no se posee ningún nivel artístico se ha decidido realizar un personaje stickman y utilizar el resto de los elementos del mundo real pasando ciertos filtros para oscurecer las imágenes. Para realizar estos recursos se ha utilizado Inkscape, Gimp y Pixlr (software libre).

Los recursos se pueden agrupar en 4 grandes bloques:

- Personaje y movimientos
- Enemigos
- Elementos decorativos y trampas
- Elementos interactivos

El personaje principal:
El personaje principal es un stickman dibujado con Inkscape lo que ha facilitado la creación de animaciones.
Se utilizará una máquina de estados con las diferentes animaciones: esperar, andar, correr y saltar (y esto también puede realizarlo con el paraguas cerrado y también con el paraguas abierto). Además el personaje puede atacar cuando el paraguas está cerrado y entreabrirlo.
A continuación se muestran algunos ejemplos de las animaciones que conforman la máquina de estados:

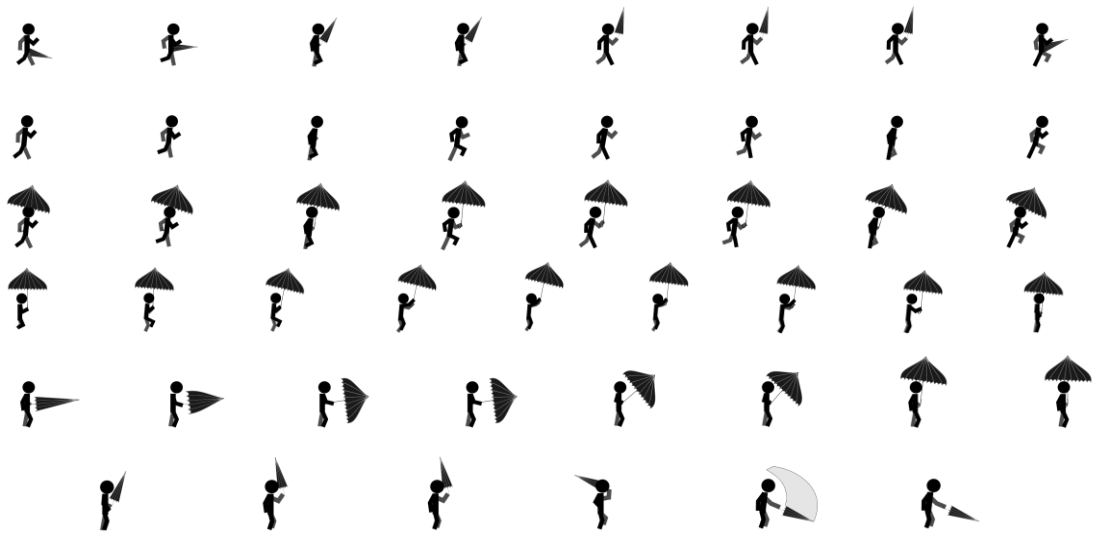


Figura 23: Personaje principal

Los enemigos:
Lobos: se ha utilizado un lobo real en movimiento parándolo en cada frame y filtrando la imagen posteriormente.

Figura 24: Enemigo Lobo

Enemigos stickman: utilizando los movimientos del personaje principal se dibujan otros stickman cambiando las armas.
A continuación se muestra la animación de ataque de un enemigo con cuchillo, un enemigo con hacha y un enemigo con guadaña.

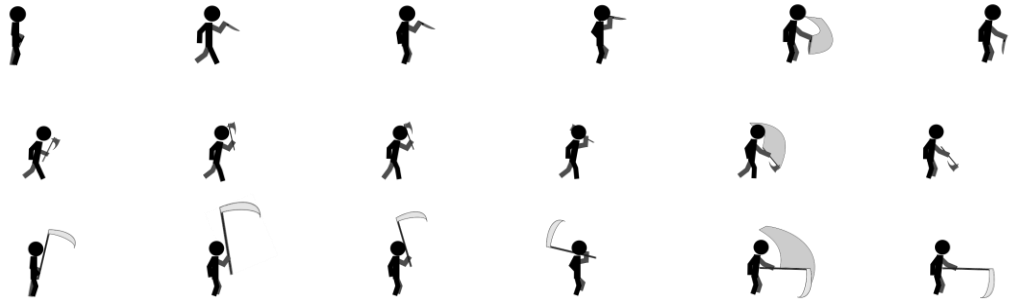


Figura 25: Enemigos stickman

Elementos decorativos:
Se utilizan fotos (realizadas por el autor y algunas con licencia creative commons) para crear los elementos decorativos y las trampas. Se han utilizado todo tipo de elementos: acantilados, hierbas, árboles, nubes… y para los fondos se han utilizado montañas, paisajes, llanuras, etc.

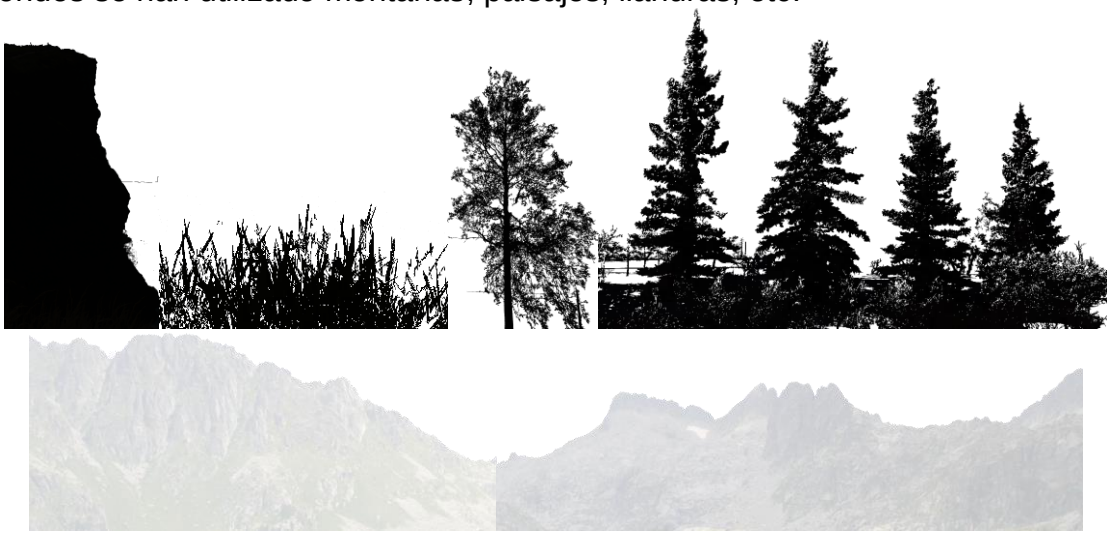


Figura 26: Elementos decorativos

Elementos interactivos:
Los elementos interactivos elegidos son la lluvia y el fuego, y se pueden realizar utilizando sistemas de partículas de Unity.

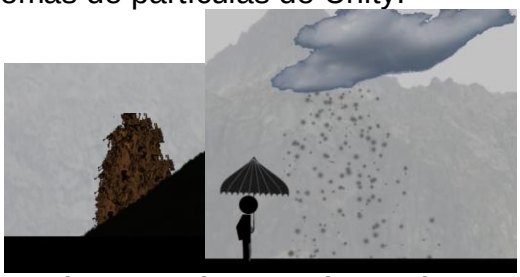


Figura 27: Elementos interactivos

3.5 Prototipo funcional

Para continuar con el diseño se ha creado un prototipo funcional en Unity. Este prototipo funcional se solapa con la parte de implementación por lo que los detalles se explicarán en dicho apartado.
El objetivo de este prototipo es hacer un “vertical slide”, un juego con ciertos niveles que represente la mecánica de Umbrellibur y que permita ver cómo va a ser en un futuro.
A continuación se muestran ciertas capturas de pantalla del prototipo funcional.

Figura 28: Prototipo alta fidelidad pantalla inicial

Figura 29: Prototipo alta fidelidad selección de nivel

Figura 30: Prototipo alta fidelidad opciones

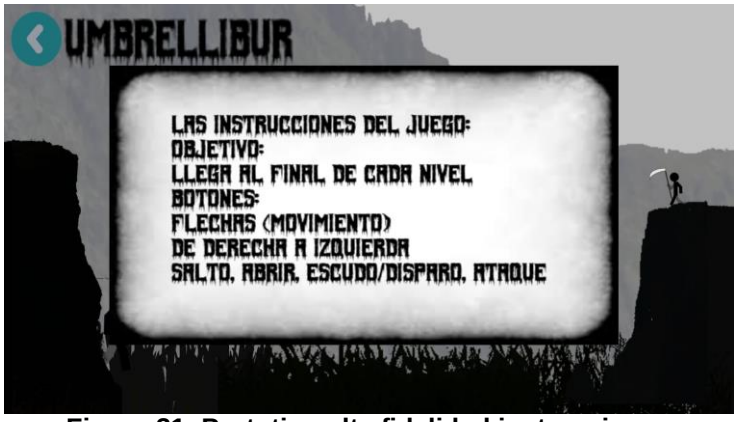


Figura 31: Prototipo alta fidelidad instrucciones

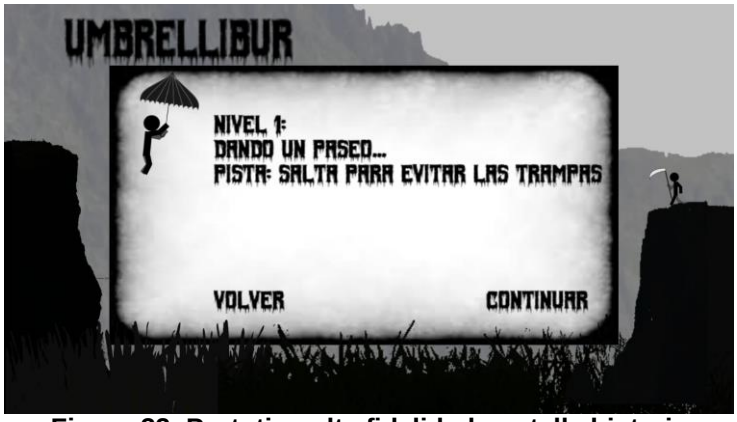


Figura 32: Prototipo alta fidelidad pantalla historia

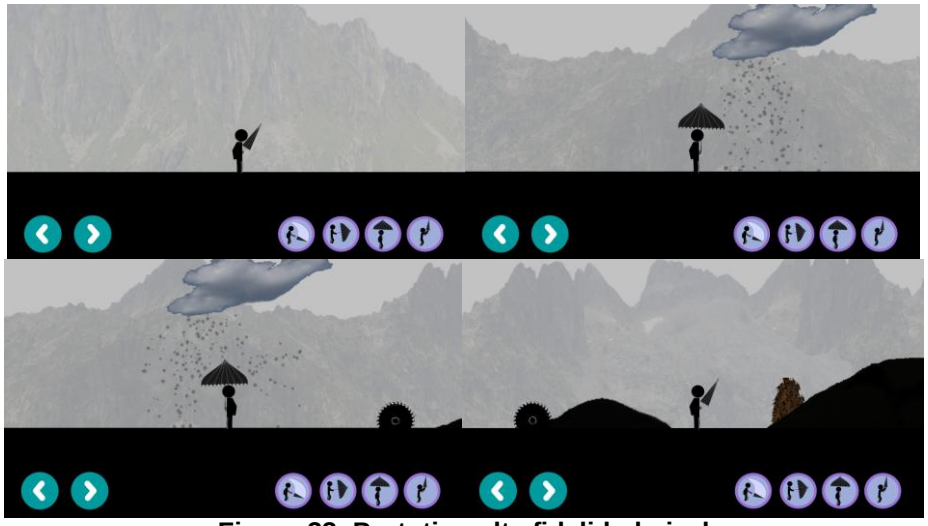


Figura 33: Prototipo alta fidelidad nivel

3.6 Evaluación del prototipo funcional

Una vez se ha desarrollado el prototipo funcional se difunde entre un grupo reducido de potenciales usuarios con el objetivo de realizar una evaluación. Una vez han probado el prototipo realizarán una encuesta planteada a través de Google Forms para evaluar el prototipo.
A continuación se muestran los resultados de la encuesta:

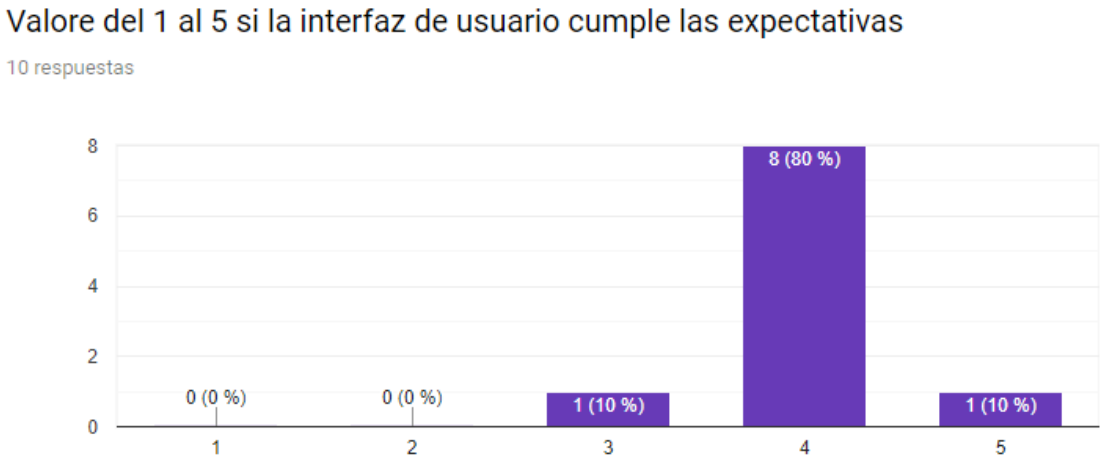


Figura 34: Valoración de la UI

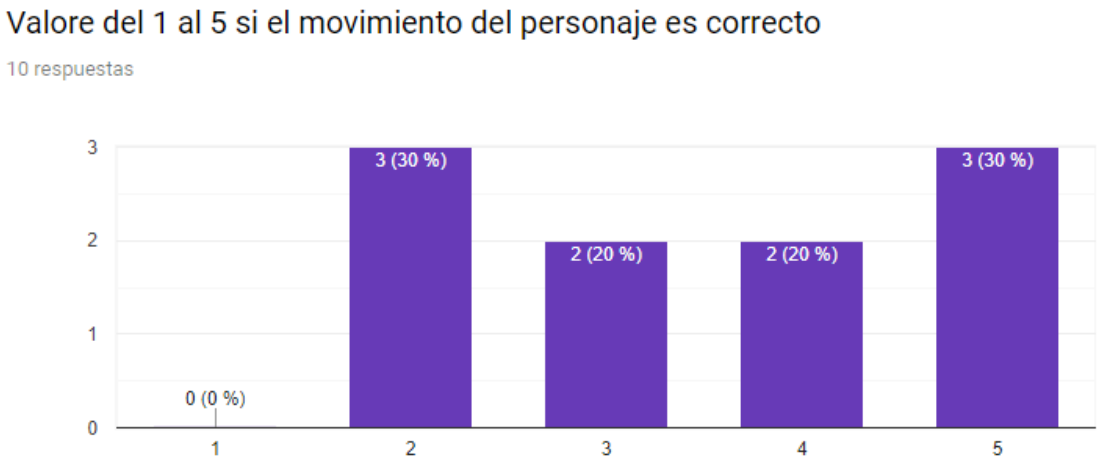


Figura 35: Valoración del movimiento

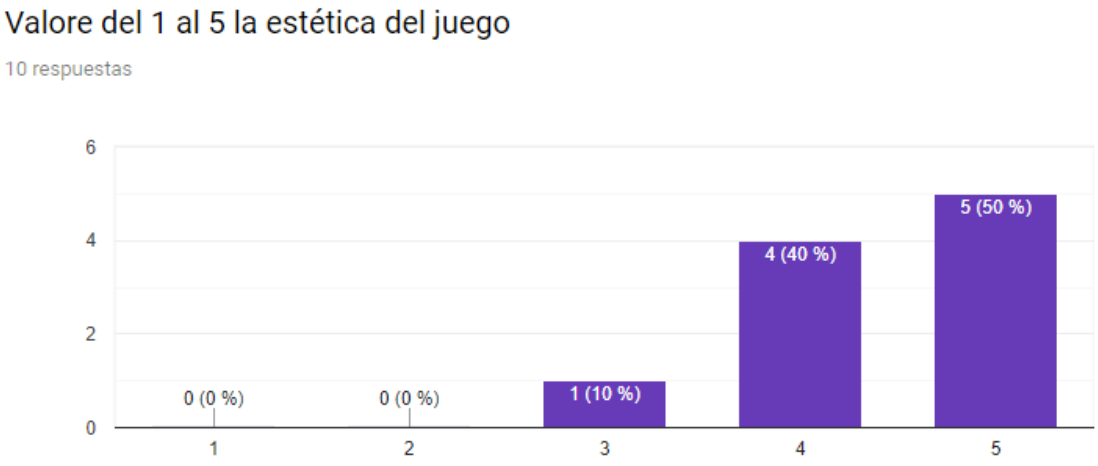


Figura 36: Valoración estética

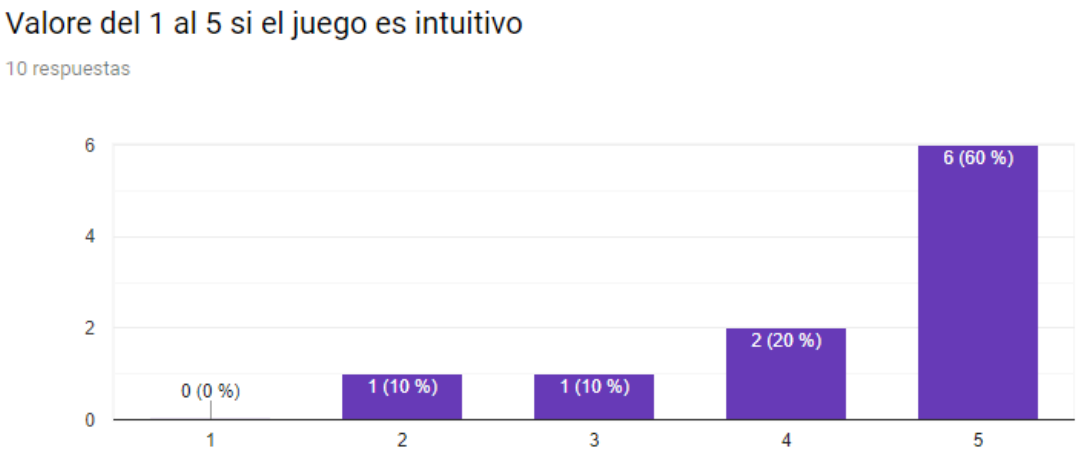


Figura 37: Valoración de si es intuitivo

Figura 38: Valoración de uso

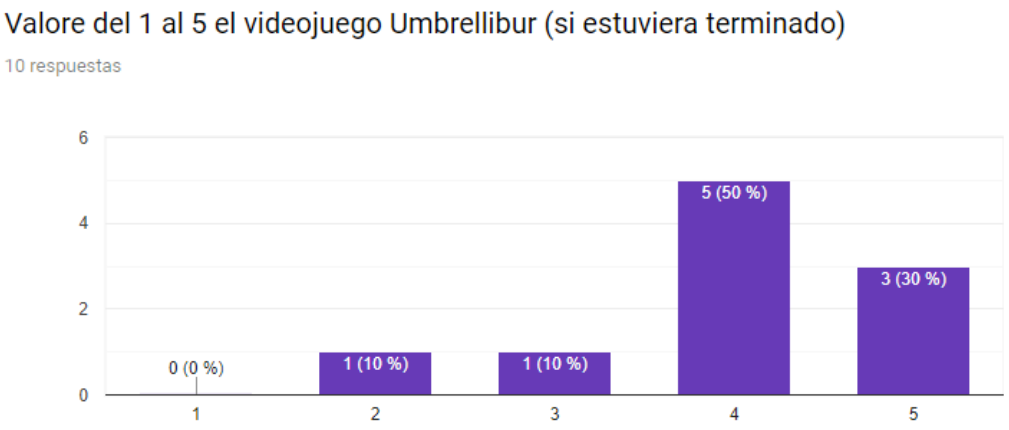


Figura 39: Valoración general

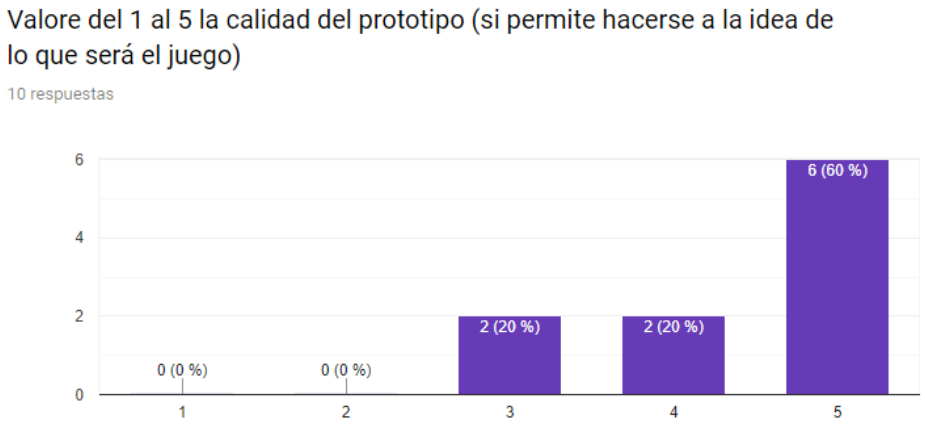


Figura 40: Valoración del prototipo

Además de estas preguntas se solicitan sugerencias y bugs. Con estos datos se podrá entender qué debe ser esencial en la fase de implementación final.
En cuanto a la interfaz de usuario, todos los participantes la consideran al menos adecuada por lo que se considera correcta y solo habrá que pulir los detalles para que haya una correcta navegación.
En cuanto al movimiento hay cierta división de opiniones. Se considera que los movimientos son correctos pero que se producen bugs molestos durante el juego.
Sin embargo, la estética recibe una valoración muy positiva (9 de 10 la consideran buena o muy buena).
En relación con si el juego es intuitivo o no, la mayor parte de los usuarios (6 de 10) lo considera muy intuitivo. Sin embargo hay casos que no lo consideran intuitivo por lo que se tratará de realizar un tutorial a lo largo de los primeros niveles para facilitar el uso correcto del juego.
Solo uno de los usuarios considera que no utilizaría el juego debido a que no utiliza juegos para dispositivos móviles (por lo que sería bastante complicado llamar su atención).
La valoración general del juego es buena (5 lo consideran bueno, 3 muy bueno y solamente 1 lo considera malo) y en cuanto al prototipo el 60% considera que es una muestra muy clara de cómo será el juego.

4. Implementación

En esta apartado se detalla la implementación del videojuego: las herramientas utilizadas, la estructura del proyecto, el código y la funcionalidad de las diferentes pantallas.

4.1 Herramientas utilizadas

La base del trabajo se realiza con Unity. Como entorno de desarrollo se utiliza Visual Studio. Este es el IDE utilizado en el proyecto debido a que es el más recomendado para desarrollar en Unity y facilita la edición y escritura de scripts en C# (un lenguaje de programación orientada a objetos).
Se plantea desde el inicio la exportación a dispositivos Android y se hace uso de un terminal Xiaomi MiA1 para realizar las pruebas.

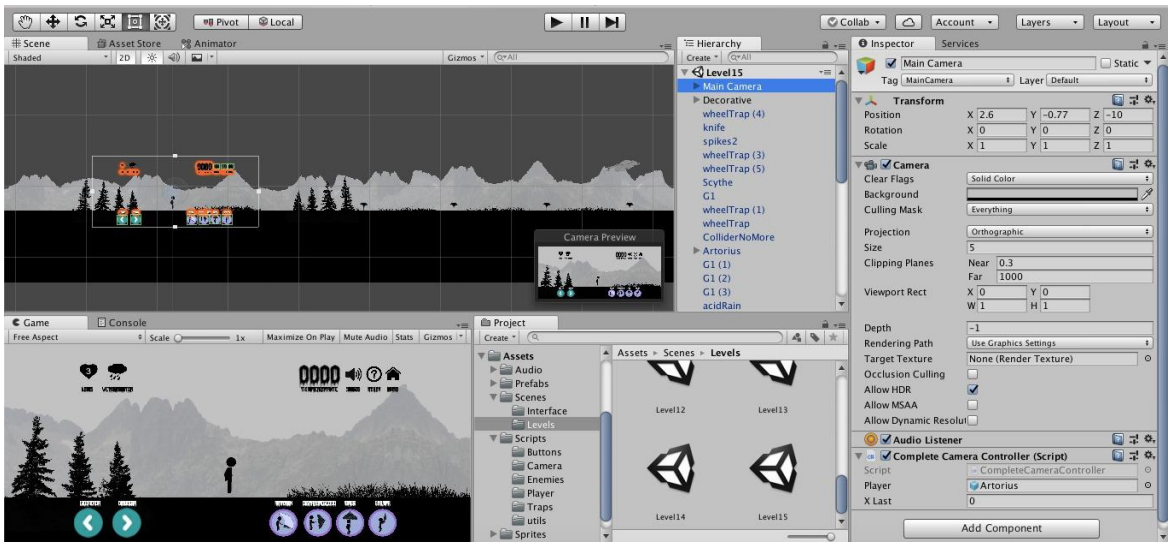


Figura 41: Unity

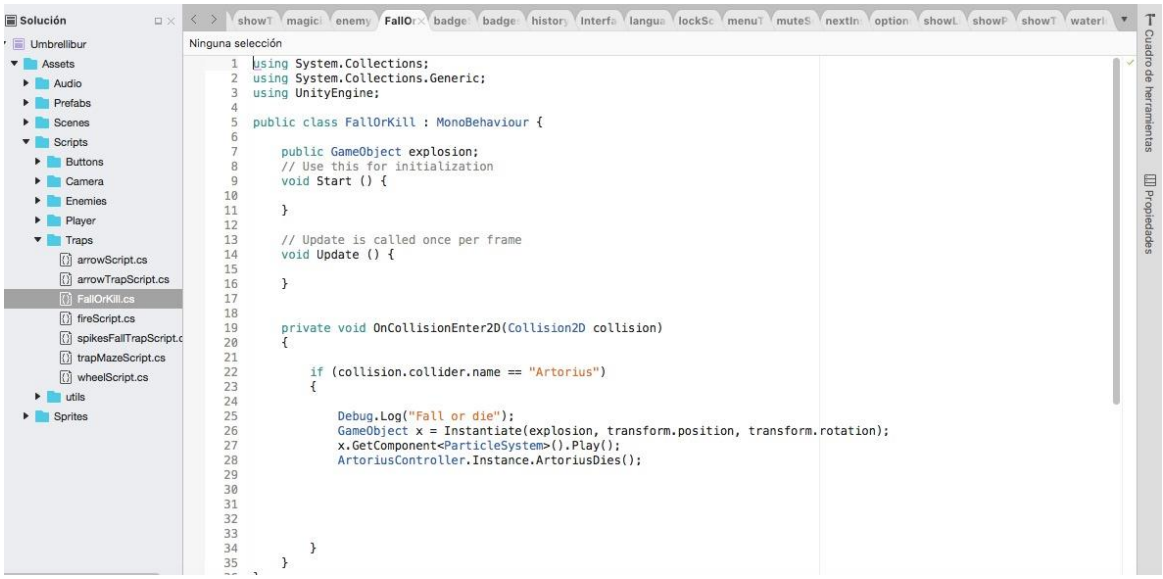


Figura 42: Visual Studio

4.2 Estructura del proyecto

Para la realización del videojuego es eficiente guardar de forma ordenada los diferentes recursos utilizados. Para ello la carpeta Assets se ha organizado de la siguiente forma:

/Audio
/Prefabs
/Scenes
/Scripts
/Sprites

De esta forma quedan separados los diferentes recursos necesarios y se pueden ubicar con rapidez.

Una vez estructurado el proyecto se debe plantear cómo se organizarán las diferentes pantallas para cumplir con el mapa de navegación especificado en el diseño.

Las pantallas se agrupan en dos carpetas de escenas (interfaz y niveles). En el siguiente apartado se explicará la implementación de las escenas.

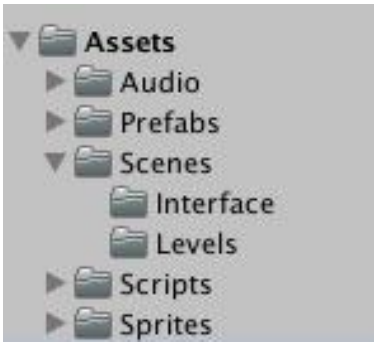


Figura 43: Estructura Assets

4.3 Implementación del videojuego

La implementación de Umbrellibur tiene dos grandes bloques. La interfaz de usuario y los niveles de juego.

4.3.1 Interfaz de usuario

La interfaz de usuario se divide en varias pantallas (que a su vez son escenas de Unity): menú principal, menú de niveles, menú de opciones, pantalla de instrucciones, pantalla de logros, pantalla de historia y los niveles en sí.
Para la realización de estas pantallas ha sido necesario programar ciertos scripts para manejar la correcta navegación entre las mismas así como para controlar la música de fondo. La música se ha seleccionado de patrickdearteaga.com y los textos son de dafont.com.

Menú principal:
En la pantalla del menú principal se muestra el título y cuatro opciones frente a un fondo que muestra la estética del videojuego. Las opciones son Jugar (que abrirá el menú con los niveles), Instrucciones, Opciones y Salir.

Figura 44: Menú de inicio

Menú de niveles:
El menú de niveles muestra un enlace a la pantalla de logros y botones para abrir los niveles desbloqueados. Se ha programado de manera que los niveles desbloqueados se muestren en negro a excepción del último desbloqueado (el que todavía no ha sido superado) que se mostrará en rojo. Los niveles bloqueados mostrarán un candado para indicar que no es posible jugarlos.

Figura 45: Menú de niveles

Pantalla de logros o insignias:
La pantalla de insignias mostrará las insignias que se han desbloqueado y las insignias que quedan por desbloquear.

Figura 46: Pantalla de logros

Menú de opciones:
El menú de opciones permite modificar 3 parámetros del juego:

- La dificultad: permite modificar el la dificultad de los enemigos del juego.
- La música: permite habilitar y deshabilitar el sonido en el juego.
- El idioma: permite elegir el idioma del juego entre español e inglés.

Figura 47: Menú de opciones

Pantalla de instrucciones:
Es una pantalla similar a la pantalla de historia en la que se explica mediante texto el funcionamiento del juego. Hay cuatro textos, que detallan el objetivo del juego, los controles y los diferentes elementos con los que interactúa el personaje del juego, entre los que se puede navegar pulsando un botón.

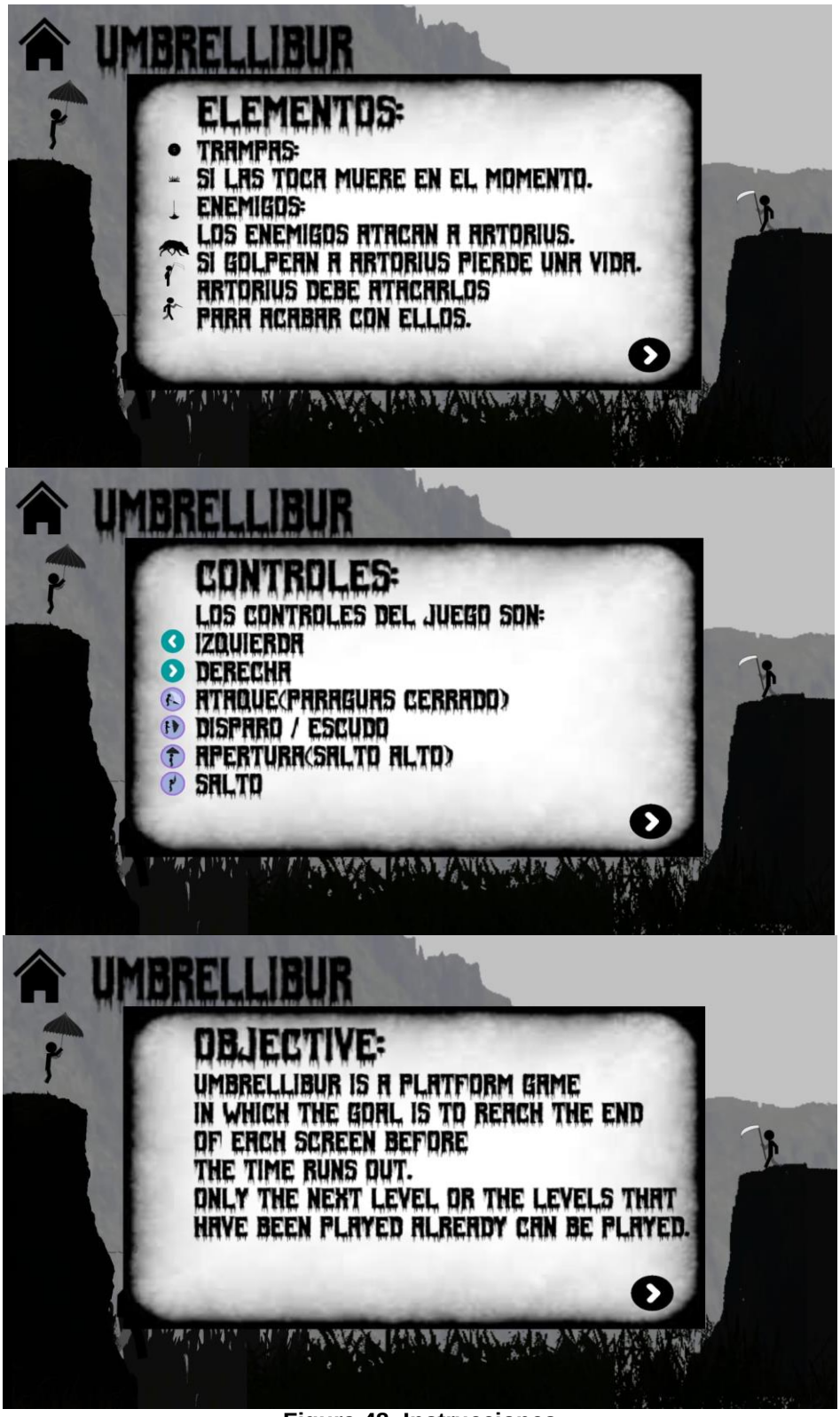


Figura 48: Instrucciones

Pantalla de historia:
Esta pantalla se muestra antes de comenzar cada nivel. En ella se va avanzando en la historia del juego. Permite volver al menú de niveles o empezar el nivel.

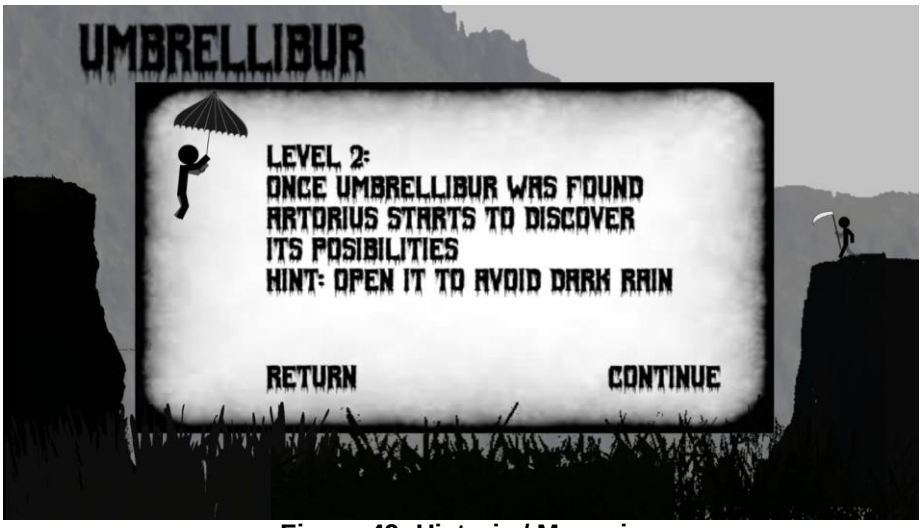


Figura 49: Historia / Mensaje

Niveles:
Los niveles tienen unos indicadores y botones en la zona superior y unos controles en la zona inferior de la pantalla. Los indicadores se encuentran en la barra superior. Muestran las vidas restantes del personaje, si tiene el paraguas cargado de agua y el tiempo disponible. Además se puede activar o desactivar el sonido, volver a inicio y mostrar la ayuda. La implementación de la funcionalidad de los niveles se detalla en el siguiente apartado.

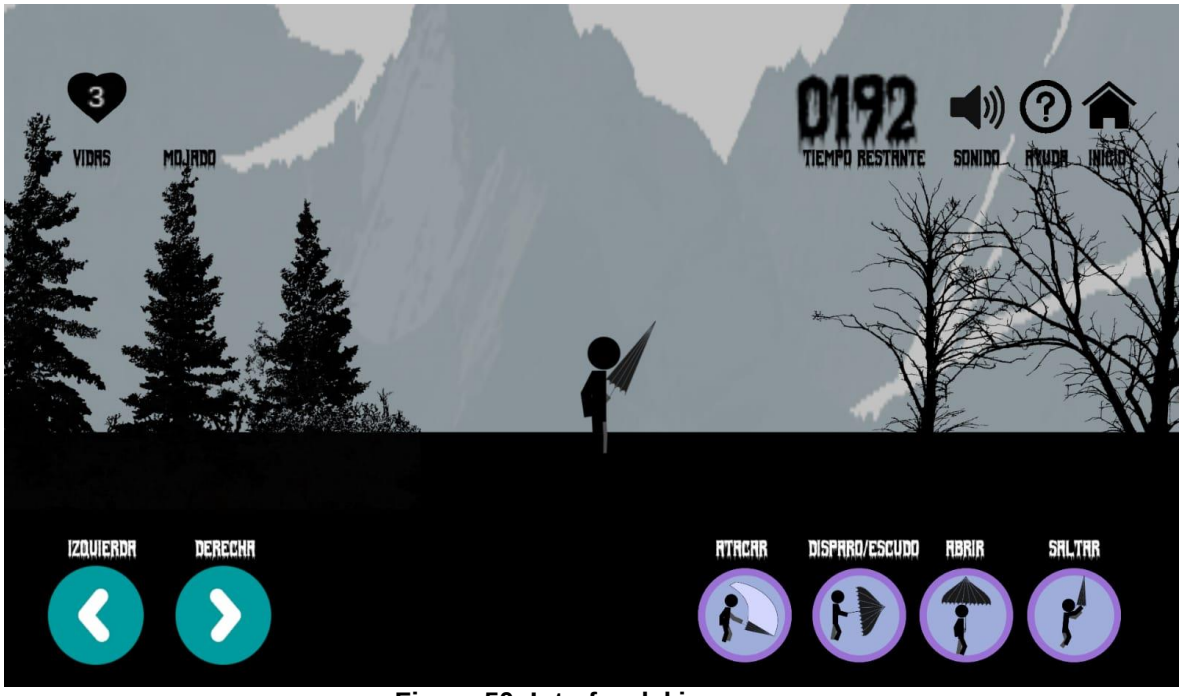


Figura 50: Interfaz del juego

4.3.2 Niveles de juego

Los niveles del juego cuentan con varios elementos que se deben mencionar:

Figura 51: Nivel de juego

Personaje principal:

El personaje principal, Artorius, es un stickman que tiene ciertos movimientos que el usuario debe elegir para sortear las trampas y terminar los niveles.
Los movimientos son los siguientes:

- Izquierda
- Derecha
- Salto
- Abrir paraguas
- Abrir paraguas en horizontal
- Ataque

El movimiento se realiza utilizando la función de Unity translate y en el caso del resto de movimientos al pulsar los botones el personaje muestra una animación diferente que se realiza en una máquina de estados. Por ejemplo, al comenzar a caminar se activa un booleano de forma que se pasa del estado “Idle” al estado “Walk”. Se utilizan los booleanos que se observan en la siguiente figura.

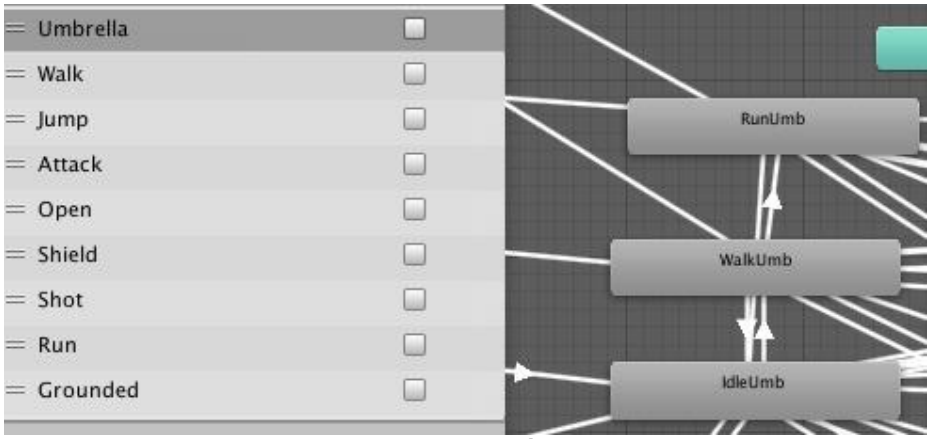


Figura 52: Fragmento de máquina de estados

Elementos del juego:
Trampas:
Las trampas matan a Artorius. Tienen un Collider (normalmente polygon collider) y al producirse una colisión con el personaje principal se llama al método que termina el juego.

Hay trampas de varios tipos:

- Pinchos: estáticos o caen del techo.
- Ruedas: se mueven en horizontal.
- Mazas y cuchillas: se balancean.
- Flechas: ciertas piedras lanzan flechas.

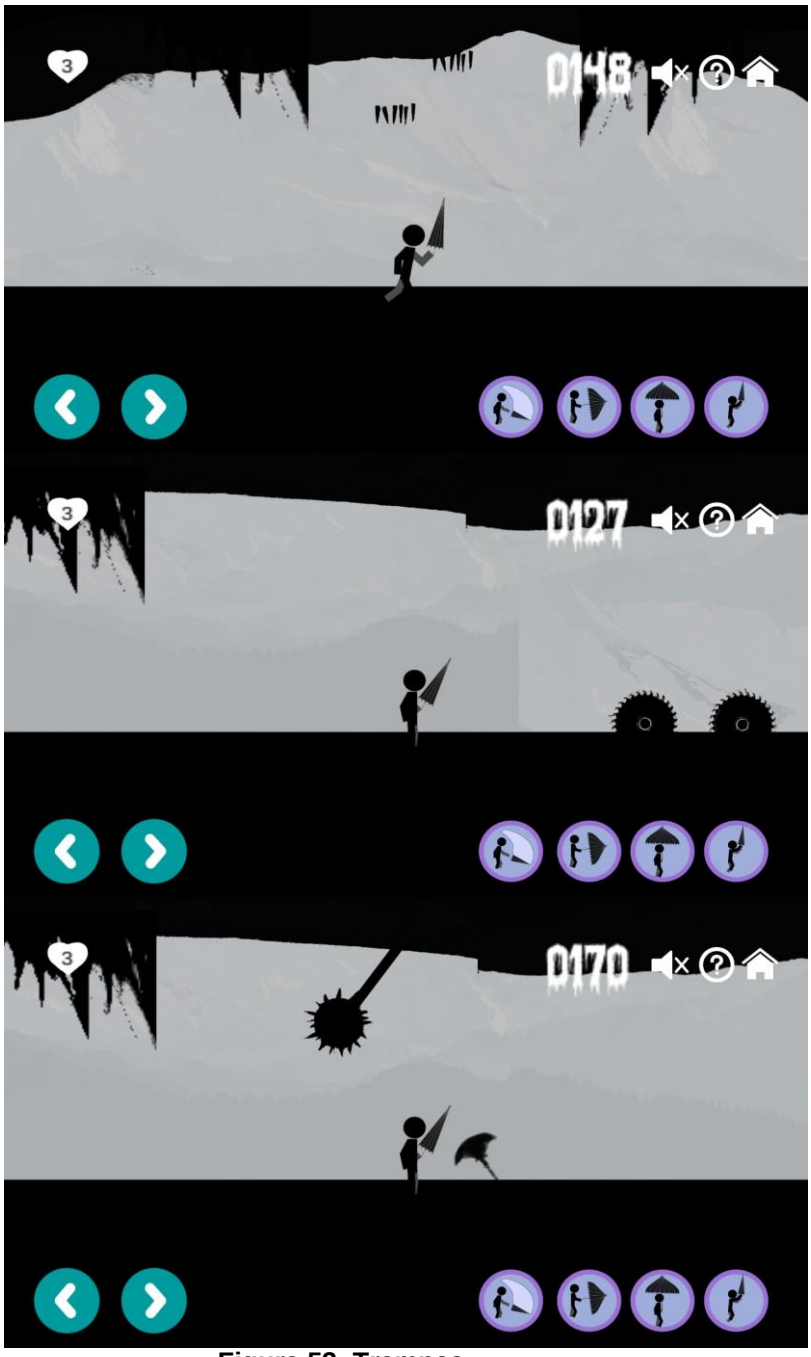


Figura 53: Trampas

Enemigos:
Persiguen y golpean a Artorius y le restan una vida. En caso de ser la última, se termina el juego.
Hay varios tipos de enemigos (Hacha, guadaña, cuchillo). Se diferencian en la distancia a la que persiguen a Artorius, la velocidad y el número de golpes que aguantan. Además, dependiendo de la dificultad seleccionada, aguantarán más o menos golpes.
El funcionamiento es similar al del personaje principal y tienen una máquina de estados reducida que cuenta con 3 estados: Idle, Attack y Run con dos booleanos (walk y attack). Si Artorius está a cierta distancia el enemigo comienza a perseguirle. Si además está cerca, tratará de golpearlo.

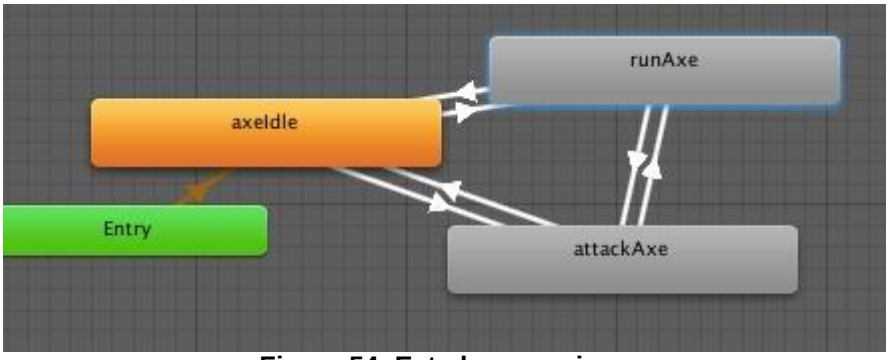


Figura 54: Estados enemigos

Figura 55: Enemigos

En el último nivel se ha añadido un enemigo final. El mago además de funcionar como un enemigo lanza hechizos que dañan al personaje.

Figura 56: Enemigo final

Elementos interactivos:
Se han programado utilizando sistemas de partículas y se han tenido en cuenta las colisiones. Hay de tres tipos:
Lluvia normal: sirve para cargar el paraguas de agua para apagar fuegos.

Figura 57: Lluvia normal

Lluvia negra: obliga a pasar con el paraguas abierto o mata a Artorius.

Figura 58: Lluvia negra

Fuego: mata a Artorius pero puede apagarse disparando agua.

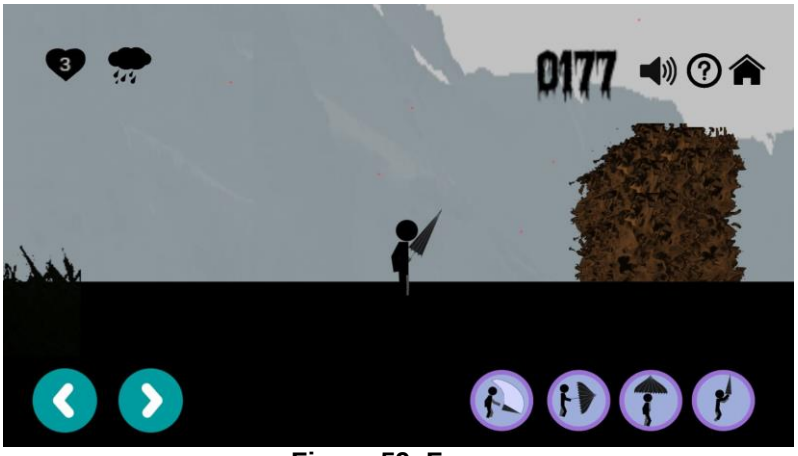


Figura 59: Fuego

Persistencia:
Para el correcto funcionamiento del juego es necesario guardar ciertos datos de forma persistente en el dispositivo. Para ello se ha utilizado la funcionalidad de Unity PlayerPrefs que permite guardar una serie de valores en un diccionario clave/valor.
En concreto se ha utilizado para guardar la configuración (Idioma, dificultad y sonido), el nivel seleccionado, el nivel máximo y para comprobar si se han superado los logros.

Creación y diseño de niveles:
Generalmente los usuarios de un videojuego no suelen leerse las instrucciones e intentan jugar lo antes posible. Para facilitar la comprensión del funcionamiento del juego los primeros 5 niveles se plantean como un tutorial en los que el usuario irá conociendo paso a paso los diferentes botones y su funcionamiento. En estos niveles la dificultad es ligeramente inferior y no suele combinarse la utilización de los botones en espacios cercanos de tiempo.
A partir del nivel 5 los niveles pretenden ser cada vez más complejos y a su vez ser cada nivel diferente del anterior con la aparición del algún elemento nuevo o simplemente combinaciones más complejas.
Por último el nivel 15 terminaría con el enemigo final que es un mago que lanza hechizos.

4.4 Principales problemas encontrados

Para la implementación del videojuego se han detectado tres grandes problemas que en el resultado final han tenido un cierto impacto.

Problema 1: Falta de conocimiento de diseño 2D
La falta de conocimiento de diseño 2D ha impedido la realización de un juego visualmente atractivo. El desconocimiento de las herramientas de edición ha provocado un uso excesivo de tiempo para poder conseguir los assets necesarios.
Además ha habido ciertos problemas con los sprites creados debido a los centros de gravedad de los mismos, etc. En definitiva, ha sido bastante complicado conseguir un movimiento que pareciera realista.

Problema 2: Falta de tiempo
Se realizó una planificación que exigía muchas horas para el desarrollo del juego. La realidad ha sido que el tiempo requerido y el dedicado no ha sido suficiente para pulir los detalles, especialmente no se tuvo en cuenta que la implementación tendría que estar terminada a principios del mes de mayo. A pesar de la falta de tiempo se ha conseguido llegar a un resultado que cumple con los objetivos iniciales.

Problema 3: Falta de conocimiento de Unity
Se parte del conocimiento adquirido en la asignatura de Introducción a videojuegos en dispositivos móviles. Esta asignatura tiene algunas carencias como por ejemplo no haber sido necesario desarrollar juegos con funcionalidades específicas de dispositivos móviles. Cada vez que se han necesitado funcionalidades específicas se ha tenido que investigar cómo realizar estas funcionalidades.

Problema 4: Estado de los equipos de desarrollo
Los equipos en los que se ha desarrollado el videojuego tienen ciertas carencias. En el caso del ordenador Windows de seis años de antigüedad ha sido imposible desarrollar y se ha utilizado para otras funciones de diseño y documentación. En el caso del Mac los problemas principales han sido el tamaño y el espacio de almacenamiento. Para desarrollar un videojuego una pantalla de 11 pulgadas es muy pequeña. También ha habido problemas de compatibilidad con Word por lo que la documentación no ha podido escribirse en este equipo.

5. Pruebas

Debido al desarrollo del proyecto se han realizado todo tipo de pruebas sin seguir un esquema fijado con antelación. Se han realizado pruebas en todo momento intentando garantizar el correcto funcionamiento del juego y de la interfaz.

Las pruebas realizadas se han centrado en los siguientes aspectos:

- Interfaz: correcto funcionamiento de la navegación entre pantallas y correcto funcionamiento de los cambios en la configuración.
- Juego:
  Movimiento del personaje: botones y animaciones.
  Elementos: interacción con enemigos, trampas, lluvia, etc.
  Niveles: posibilidad de superar los niveles y cambios al superarlos.

A continuación se muestran una serie de pruebas realizadas una vez terminada la implementación.

Parte de la aplicación	**Prueba**	**Resultado esperado**	**Resultado**
Menú	Inicio	La aplicación se inicia cargando el menú principal	Correcto
Menú	Jugar	Abre el menú de niveles al pulsar el botón Jugar	Correcto
Menú	Opciones	Abre el menú de opciones al pulsar el botón Opciones	Correcto
Menú	Instrucciones	Abre la pantalla de instrucciones al pulsar el botón Instrucciones	Correcto
Menú	Salir	La aplicación se cierra al pulsar Salir	Correcto
Instrucciones	Botón avanzar	Muestra el siguiente texto de instrucciones o el primero en caso de haber terminado	Correcto
Opciones	Mute	Quita el sonido de la aplicación	Correcto
Opciones	Dificultad	Cambia la dificultad del juego y los enemigos aguantan más o menos golpes	Correcto
Opciones	Idioma	Cambia el idioma de los textos	Correcto
Menú de niveles	Candados	No permite entrar en los niveles bloqueados y se desbloquean al pasar un nivel	Correcto
Menú de niveles	Logros	Abre la pantalla de logros al pulsar el icono de Logros	Correcto

Mensaje	Volver	Vuelve al menú de niveles al pulsar el botón de Volver	Correcto
Mensaje	Continuar	Inicia el nivel correspondiente al pulsar el botón Continuar	Correcto
Menú, menú de niveles, opciones instrucciones…	Música	La música de fondo se mantiene hasta llegar a la pantalla de Historia/Mensaje	Correcto
Juego	Indicador tiempo	Baja el tiempo, se termina el juego si llega el contador a cero, cambia de valor dependiendo de la dificultad	Correcto
Juego	Movimiento	Se mueve correctamente en combinación de botones	Correcto
Juego	Indicador lluvia	Aparece el indicador si se pasa por la lluvia normal con el paraguas abierto	Correcto
Juego	Trampas	Matan a Artorius al tocarlo	Correcto
Juego	Enemigos	Golpean a Artorius y restan vidas, se mueven a diferentes velocidades y cambian dependiendo de la dificultad del juego	Parcial: El funcionamiento es correcto pero en ocasiones se producen dobles colisiones o el enemigo no llega a golpear correctamente.
Juego	Indicador vidas	Resta las vidas correctamente	Parcial: En ocasiones resta dos vidas si se produce una doble colisión.
Juego	Terminar nivel	Al llegar al final del nivel se permite pasar al siguiente y se desbloquea si es preciso	Correcto
Juego	Dificultad	Posibilidad de superar todos los niveles en todas las dificultades	Correcto

Una vez terminada la versión Beta del juego se debe realizar una serie de pruebas de validación con usuarios reales. Para ello se realizará una subida al Market del juego y se tratará de analizar y resolver los principales defectos del juego.

6. Publicación

En este apartado se detalla el proceso seguido para publicar la aplicación en el Market de Google.

6.1 Cuenta de Play Store

Para crear la cuenta de Play Store hay que seguir los pasos de la página web en los que hay que utilizar una cuenta de google, aceptar el acuerdo de desarrolladores, pagar la cuota y proporcionar ciertos datos como un correo electrónico y un número de teléfono.

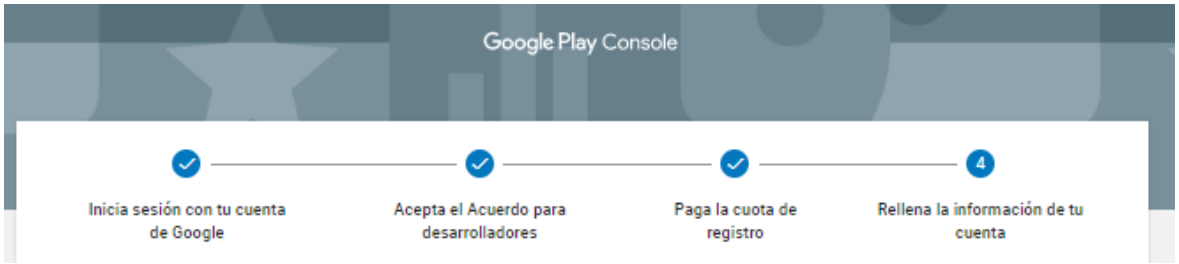


Figura 60: Fases creación Play Store

6.2 Publicación de la versión beta

Para publicar la aplicación es necesario elegir si será una versión Alfa, Beta o en producción y en este caso se ha elegido una versión Beta con un máximo de 1000 usuarios.
Para la publicación es necesario rellenar los apartados que se muestran con un icono en la siguiente figura.

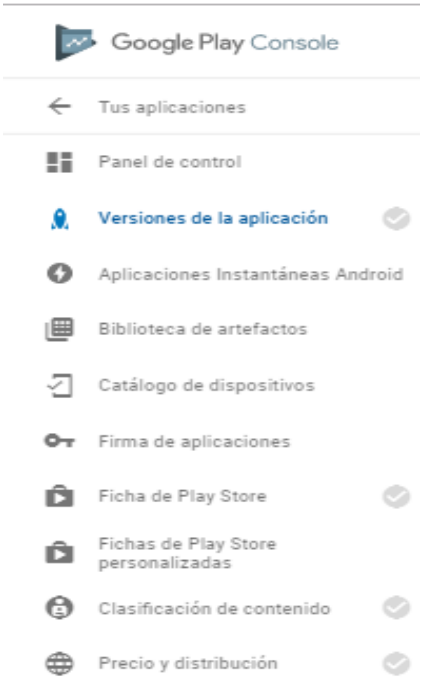


Figura 61: Consola Google Play

- Versiones de la aplicación: es necesario generar la APK (en este caso a partir de agosto de 2019 se va a restringir el formato de la APK y se ha debido modificar el proceso de generación del APK de Unity incluyendo el NDK de Android en la generación de la misma).
- Clasificación del contenido: se ha clasificado el contenido como un videojuego de plataformas rellenando una encuesta.
- Ficha: en la ficha de la aplicación se han tenido que incluir captures de pantalla, un icono y la descripción del juego (en la que se ha citado a los autores de la música y las fuentes).
- Precios y distribución: la aplicación se distribuye de manera gratuita y para todos los países disponibles.

6.3 Aplicación publicada

A continuación se puede observar el resultado final de la publicación en el Market de Google.

Figura 62: Ficha Play Store

7. Conclusiones

En este apartado se exponen las conclusiones extraídas de la realización del trabajo.

7.1 Lecciones aprendidas

A lo largo del desarrollo de la aplicación se han aprendido una serie de lecciones que permitirán en un futuro desarrollar aplicaciones con otro punto de vista. En primer lugar se ha aprendido a valorar el trabajo en equipo y en buenas condiciones. El resultado de un equipo en el que cada uno aporte lo que ha aprendido en su formación será mejor que intentar hacer un proyecto solo una persona. En segundo lugar se destaca la importancia de la motivación y del tiempo necesario para la realización de proyectos propios. Sin una alta motivación inicial este proyecto no habría salido adelante debido a la falta de tiempo y a los contratiempos surgidos durante la implementación.
Por último se ha visto la necesidad de contar con unos equipos de trabajo modernos para el desarrollo. En este caso los equipos con los que se contaba y la conexión a internet han limitado el desarrollo y sobre todo han generado momentos de frustración.

7.2 Análisis de la metodología del trabajo

La metodología utilizada ha sido una metodología en cascada en la que se divide el trabajo en fases: análisis, diseño, implementación y pruebas. Además se ha utilizado Scrum para subdividir estas fases en Sprints que han coincidido con las entregas de las PEC. Esto ha permitido cambiar ligeramente el análisis inicial y hacer una aplicación completa y adaptada a los cambios surgidos durante el proceso.
El seguimiento de la metodología ha sido correcto aunque en ciertas fases se ha tenido que dedicar más tiempo del planificado especialmente en la fase de diseño e implementación. La fase final de pruebas se ha visto reducida ya que se han realizado pruebas a lo largo de todo el desarrollo.

7.3 Cumplimiento de objetivos

Los objetivos del trabajo detallados en la introducción se han cumplido. Se ha desarrollado una planificación realista del proyecto y se han puesto en práctica los conocimientos adquiridos en el máster de forma que se ha realizado la aplicación correctamente. De esta forma se ha adquirido la experiencia suficiente para poder afrontar proyectos en un futuro.
En cuanto al cumplimiento de los requisitos funcionales planteados, se han cumplido correctamente todos los requisitos planteados como mínimos y deseados. En el caso de los requisitos opcionales se han integrado los logros y se ha realizado la subida al Market (de gran importancia debido a que no se había realizado a lo largo del máster).
Los requisitos no funcionales tienen un carácter más subjetivo debido a la valoración entre otros de la estética. A pesar de esto se considera que se ha

llegado a un cumplimiento correcto teniendo en cuenta que no se tenía formación en este ámbito.

7.4 Líneas de trabajo futuro

Aunque el resultado de la implementación es satisfactorio existen múltiples líneas de trabajo futuro tanto para mejorar el juego como para hacer el juego más atractivo a los usuarios o monetizarlo.
Unos ejemplos serían:

- Mejorar la estética del juego con ayuda de un diseñador gráfico.
- Añadir más elementos al juego (nuevos enemigos, trampas, puzzles...)
- Mejorar, alterar o añadir movimientos en función de las valoraciones y comentarios de los usuarios.
- Más niveles o mundos. Por ejemplo, se podría crear otro juego similar en otro ambiente (como en el siglo XXI).
- Añadir más logros.
- Añadir modos de multijugador (cooperativo o competición).
- Monetización (publicidad, coste o poner un número de vidas por hora y pago por vida).
- Guardado del estado de la partida en la nube (para los jugadores que hayan perdido el móvil o borrado la App).
- Exportación a IOS.

8. Glosario

Apk: Android Package (APK) es el formato de archivo del paquete utilizado por el sistema operativo Android para la distribución e instalación de aplicaciones móviles y middleware.

Colliders: son componentes que definen la forma de un objeto para las colisiones físicas.

Firebase: es una plataforma de Google de desarrollo móvil en la nube.

IDE: es una aplicación informática que proporciona servicios integrales para facilitar el desarrollo de software para el programador.

Idle: es la animación que define la inactividad.

Máquina de estado: es un modelo de comportamiento de un sistema con entradas y salidas, donde las salidas dependen no solo de las señales de entrada actuales, sino también de las anteriores.

Market: es una plataforma para la distribución digital de aplicaciones móviles para dispositivos móviles. Google Play Store es un ejemplo para aplicaciones del sistema operativo Android.

Playerprefs: Almacena y accede a las preferencias del jugador entre las sesiones del juego.

Prefabs: El sistema Unity Prefabs permite crear, configurar y almacenar un GameObject completo con todos sus componentes, valores de propiedad e hijos como un activo reutilizable.

Singleton: es un patrón de diseño que le permite limitar la creación de objetos que pertenecen a una clase o el valor de un tipo a un solo objeto. Su intención es asegurar que una clase tenga solo una instancia y proporcionarle un punto de acceso global.

Stickman: es un personaje hecho con trazo simple. Un cuerpo, dos brazos, dos piernas y una cabeza redonda.

UI: la interfaz de usuario es todo lo que está diseñado en un dispositivo de información con el que una persona puede interactuar.

Unity: es un motor de desarrollo diseñado para la programación de videojuegos.

9. Bibliografía

Referencias

[1] Pérez Martín, J., García de Diego Martínez, A., & Parra Valcarce, D. (2009). Evolución tecnológica e impacto laboral y socio-económico de los videojuegos. *Revista Internacional de Comunicación Audiovisual, Publicidad y Literatura, 1 (7), 167-176.*

[2] Duch i Gavaldà, J., & Tejedor Navarro, H. (2011). Introducción a los videojuegos, septiembre 2011.

[3] Belli, S., & López Raventós, C. (2008). Breve historia de los videojuegos. *Athenea Digital: revista de pensamiento e investigación social*, (14), 159-179.

[4] Parra Valcarce, D., García de Diego Martínez, A., & Pérez Martín, J. (2009). Hábitos de uso de los videojuegos en España entre los mayores de 35 años.

[5] Aguado, J. M., Martínez, I. J., & Cañete-Sanz, L. (2015). Tendencias evolutivas del contenido digital en aplicaciones móviles. *El profesional de la información*, *24*(6), 787-795.

[6] Mantilla, M. C. G., Ariza, L. L. C., & Delgado, B. M. (2014). Metodología para el desarrollo de aplicaciones móviles. *Tecnura: Tecnología y Cultura Afirmando el Conocimiento*, *18*(40), 20-35.

[7] Enriquez, J. G., & Casas, S. I. (2014). Usabilidad en aplicaciones móviles. *Informes Científicos-Técnicos UNPA*, *5*(2), 25-47.

Aplicaciones de google play

https://play.google.com/store/apps/details?id=com.ubj.typoman, 08/05/2019.
https://play.google.com/store/apps/details?id=com.ubisoft.raymanadventures, 08/05/2019.
https://play.google.com/store/apps/details?id=com.FDGEntertainment.redball4.gp , 08/05/2019.
https://play.google.com/store/apps/details?id=com.djinnworks.ss18, 08/05/2019.
https://play.google.com/store/apps/details?id=com.GabrielLimberger.StickmanParkourPlatform, 08/05/2019.
https://play.google.com/store/apps/details?id=com.revi.umbrellafalling, 08/05/2019.
https://play.google.com/store/apps/details?id=nitin.sangale.androidgames.umbrellamaster, 08/05/2019.
https://play.google.com/store/apps/details?id=com.PPM.Umbrellibur, 08/05/2019.

Webgrafía (herramientas utilizadas)

https://docs.unity3d.com, 08/05/2019.
https://search.creativecommons.org/, 08/05/2019.
https://www.dafont.com, 08/05/2019.
https://www.creartuavatar.com/index.php, 08/05/2019.
https://wireframe.cc/, 08/05/2019.
https://patrickdearteaga.com/es/musica-libre-derechos-gratis/, 08/05/2019.
https://convertio.co/es/ogg-mp3/, 08/05/2019.