

Desarrollo de un videojuego de tipo Idle con elementos de rol con Unity.

Alumno: Raúl Campos Guinda

Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles
Desarrollo de videojuegos con Unity

Consultor: Pau Dominkovics Coll

Responsable de la asignatura: Carles Garigues Olivella

27 de Mayo del 2021



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

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Desarrollo de un videojuego de tipo Idle con elementos de rol con Unity. (Nombre a definir)</i>
Nombre del autor:	<i>Raúl Campos Guinda</i>
Nombre del consultor/a:	<i>Pau Dominkovics Coll</i>
Nombre del PRA:	<i>Carles Garigues Olivella</i>
Fecha de entrega (mm/aaaa):	03/2021
Titulación:	<i>Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles</i>
Área del Trabajo Final:	Desarrollo de videojuegos con Unity
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>Videojuego, Juego, Unity</i>
Resumen del Trabajo:	
<p>En este TFM se encuentra la documentación acerca del desarrollo de un videojuego de tipo clicker o de inactividad para dispositivos Android utilizando el motor Unity. Este formato de juegos proporciona un componente de adicción y ha ganado popularidad en los últimos años. Añadiendo un elemento diferenciador a un videojuego de este tipo, será posible encontrar un hueco en el mercado actual, generando una oportunidad de negocio. En el juego, se llevará a cabo una progresión constante eliminando enemigos cada vez más peligrosos. Para conseguirlo, mejoraremos a nuestro personaje con diferentes mecánicas características de juegos de rol. El objetivo final es obtener un juego que consiga ser adictivo y mantenga la atención del público con mecánicas originales.</p>	
Abstract:	
<p>In this Masters Project you will find the documentation about the development of a clicker or idle videogame for Android devices using the Unity engine. This gaming format provides an element of addiction and has recently become popular. By providing this videogame with a differentiating element, it would be possible to find a niche in the current market, creating a business opportunity. In the game, a constant progression will take place eliminating increasingly dangerous enemies. To achieve this, we will improve our character with different mechanics characteristic of role-playing games. The ultimate goal is to get a game that manages to be addictive and keeps the audience's attention with original mechanics.</p>	

Índice

1.	Introducción	1
1.1.	Contexto y justificación del Trabajo	1
1.2.	Objetivos del Trabajo	2
1.3.	Enfoque y método seguido.....	3
1.4.	Planificación del Trabajo	4
1.5.	Breve resumen de productos obtenidos	7
1.6.	Breve descripción de los otros capítulos de la memoria	7
2.	Diseño.....	8
2.1.	Conceptualización.....	8
2.1.1.	Investigación y requisitos de usuario.....	8
2.1.2.	Definición de funcionalidades.....	10
2.1.3.	Análisis y recolección de datos en el mercado actual.....	12
2.2.	Arquitectura y Wireframes	14
2.2.1.	Planteamientos iniciales.....	14
2.2.2.	Árbol de navegación	16
2.2.3.	Bocetos a mano	16
2.2.4.	Prototipos digitales.....	19
2.2.5.	Prototipo en alta definición	23
3.	Implementación.....	24
3.1.	Desarrollo.....	24
3.2.	Recursos.....	31
3.3.	Pruebas.....	33
4.	Cumplimentación de objetivos	34
5.	Conclusiones	35
6.	Glosario	36
7.	Bibliografía.....	37
8.	Anexos	39
8.1.	Contenido del archivo de enemigos enemies.json.....	39
8.2.	Manual de instalación.....	39
8.2.1.	Abrir el proyecto en Unity.....	39
8.2.2.	Ejecutar el proyecto en un dispositivo móvil.....	39
8.2.3.	Ejecutar el proyecto a través de Google Play.	40
8.3.	Manual de usuario.....	40

Lista de figuras

Ilustración 1. Diagrama de Gantt del TFM	6
Ilustración 2. Ficha de persona: Jugador experimentado.	9
Ilustración 3. Ficha de persona: Jugador casual. ¹¹	9
Ilustración 4. Pantalla principal de Cookie Clickers™	12
Ilustración 5. Pantalla de equipamiento de Eternium.	13
Ilustración 6. Pantalla principal de Tap Titans 2.	14
Ilustración 7. Árbol de navegación.	16
Ilustración 8. Boceto a mano de la pantalla principal: diseño alternativo.	17
Ilustración 9. Boceto a mano de la pantalla principal.	17
Ilustración 10. Boceto a mano de la pantalla de equipo.	18
Ilustración 11. Prototipo digital de la pantalla principal.	19
Ilustración 12. Prototipo digital de la pantalla de atributos.	20
Ilustración 13. Prototipo digital de la pantalla de habilidades.	21
Ilustración 14. Antigua ventana de talentos en World of Warcraft.	21
Ilustración 15. Prototipo digital de la pantalla de equipo.	22
Ilustración 16. Prototipo en alta definición de la pantalla principal.	23
Ilustración 17. Elección de la plataforma de ejecución.	24
Ilustración 18. Interfaz de usuario de la pantalla principal en proporción de 16:9.	25
Ilustración 19. Interfaz de usuario de la pantalla principal en proporción de 18:9.	25
Ilustración 20. Pantalla principal con enemigo y barras de salud.	26
Ilustración 21. Pantalla de equipamiento completa.	28
Ilustración 22. Gráfica de velocidad de ataque.	29
Ilustración 23. Gráfica de la experiencia máxima por nivel.	29
Ilustración 24. Gráfica de daño absorbido por defensa.	30
Ilustración 25. Gráfica de Probabilidad de Crítico	30
Ilustración 26. Pantalla de talentos.	31
Ilustración 27. Ejemplo de elementos UI utilizados en la aplicación.	32
Ilustración 28. Ejemplo de imágenes de enemigos utilizados en la aplicación.	32
Ilustración 29. Imágenes de ejemplo de objetos utilizado en la aplicación.	33

Lista de tablas

Tabla 1. Recursos Software.....	4
Tabla 2. Recursos Hardware.	4

1. Introducción

1.1. Contexto y justificación del Trabajo

En el mercado de aplicaciones Android (Play Store) se encuentran cosechando cada vez más seguidores los videojuegos de tipo 'idle' o inactivo. Esta clase de juegos tratan de mantener una progresión continua incluso en los momentos en los que el usuario no tenga la app abierta, haciendo más atractiva la vuelta a la aplicación, para recibir ganancias y acelerar la progresión de manera exponencial. Por ejemplo, Idle Farming Empire¹, Tap Titans 2² o Cookie ClickersTM³ son aplicaciones de este formato que han conseguido más de 10 millones de descargas, pero hay muchas más. [\[1\]](#)

Dentro de este tipo de apps existen una variedad de mecánicas para conseguir una progresión dinámica, así como un abanico de temáticas diferentes para conseguir captar a los usuarios. Por lo general, los juegos 'idle' suelen tratar acerca de la recolección de un recurso u objeto, que finalmente se convierte en un incremento exponencial del mismo. Otros, en cambio, tratan de realizar una progresión enfrentándose a enemigos cada vez más fuertes, en este caso, lo que se recolecta es la fuerza con la que poder eliminarlos. Si mencionamos los ejemplos anteriores, en Idle Farming Empire controlaremos un imperio agrícola con el que podremos conseguir más y más dinero, en Tap Titans 2 controlaremos a un grupo de héroes que se irán desarrollando para lidiar con enemigos cada vez más poderosos y en Cookie Clickers el objetivo es tan sencillo como recolectar tantas galletas como sea posible. [\[2\]](#) [\[3\]](#)

Un factor común que tienen este tipo de juegos, aunque no todos, es la posibilidad de reiniciar y empezar de cero, pero partiendo con algún tipo de ventaja en la que 'renacer', aportando más reusabilidad y tiempo de vida a la aplicación. Sin embargo, la gran mayoría de estos juegos no cuentan con la suficiente profundidad para que un usuario pueda realizar sus propias decisiones para conseguir la mayor progresión posible (a su criterio) y se limitan a mantener un crecimiento rígido y monótono, donde las acciones a tomar son muy escasas, y, por tanto, haciendo que el usuario termine dejando de lado la aplicación.

También cabe destacar que existe una enorme competencia en este tipo de juegos, ya que en el mercado de aplicaciones se encuentra una ingente cantidad de juegos tipo idle, aunque muchos de ellos carecen de originalidad y mecánicas interesantes. Por esta razón se hace necesario adquirir un elemento diferenciador a la hora de desarrollar un videojuego de tipo idle para dispositivos móviles.

Por otro lado, también existe un amplio mercado en los videojuegos denominados RPG (role-playing game) con una cifra considerable de usuarios activos, en el que los jugadores irán mejorando su personaje a lo largo de una historia, misiones, batallas, etc. No obstante, en dispositivos móviles, estos tipos de aplicaciones tienden a tener una jugabilidad muy cerrada, muchos de ellos simplemente se limitan a seguir una sucesión de misiones en las que el usuario prácticamente no interactúa, ya que el control del

¹ Futureplay (Videojuego): [Link de la aplicación en Play Store](#)

² Game Hive Corporation (Videojuego): [Link de la aplicación en Play Store](#)

³ Tiny Games Srl (Videojuego): [Link de la aplicación en Play Store](#)

personaje se automatiza. Un ejemplo de una aplicación de este tipo que está triunfando actualmente es Black Desert Mobile⁴.

En otras plataformas (consolas, ordenador) es más habitual encontrar juegos de rol con más libertad de elección y personalización del personaje, provocando satisfacción en el usuario. En especial, algunos juegos que cuentan con una profunda y trabajada personalización del personaje, como ocurre en Path of Exile⁵, donde el usuario dispone de una inmensa cantidad de opciones para conseguir un personaje único a su elección.

Por todo ello, encontramos en los juegos tipo idle una falta de libertad para que el usuario puede tomar su propio camino a la hora de mejorar a su personaje, mientras que en otro tipo de juegos existen un sinnúmero de posibilidades en las que realizar una progresión del personaje. Es precisamente en este punto flaco en el que podremos aprovechar la oportunidad: crear una app de Android que sea un videojuego de tipo idle (o inactivo) con la suficiente profundidad para que el usuario pueda tomar un camino a su elección con el que obtener la mayor progresión posible. Se trata entonces de agregar elementos de rol donde el usuario sea capaz de elegir su propio camino en la progresión del personaje.

De esta manera, conseguiremos una aplicación que tenga el gigante seguimiento de usuarios como es el de tipo 'idle' y, además, aportar un elemento diferenciador que otorgue atracción al producto y aumente el tiempo de vida de la aplicación en sus dispositivos.

1.2. Objetivos del Trabajo

En este trabajo de fin de máster se propondrá y ejecutará el desarrollo de una aplicación Android que sea un videojuego 'idle' que cuente con opciones personalizables de progresión. Para ello, podemos desglosar nuestros objetivos en dos partes fundamentales: objetivos generales del propio proyecto para el desarrollo de la aplicación y objetivos funcionales del videojuego.

- Objetivos del proyecto:
 1. Realizar una planificación del proceso de desarrollo del videojuego.
 2. Documentar todo el proceso de planificación, diseño y desarrollo, en este TFM (trabajo fin de máster).
 3. Obtener un videojuego de tipo inactivo que cuente con un elemento diferenciador para captar la atención del usuario.
 4. Realizar pruebas, analizar el resultado y proponer mejoras de futuro.

- Objetivos funcionales del videojuego:

⁴ PEARL ABYSS (Videojuego): [Link de la aplicación en Play Store](#)

⁵ Grinding Gear Games (Videojuego): [Link de la página oficial](#)

1. Como elemento indispensable en este tipo de videojuegos, obtener una interfaz de usuario bien estructurada, limpia y elegante.
2. Desarrollar un sistema de juego 'idle' que funcione bien a lo largo del tiempo y aumente gradualmente la dificultad de progresión.
3. Desarrollar mecánicas para que el usuario tenga libertad de elegir sus propias elecciones para que su progresión sea dinámica y entretenida.

Finalmente, y como objetivo personal, espero asimilar nuevos conceptos en el entorno de desarrollos de videojuegos con Unity, y su exportación al sistema Android.

1.3. Enfoque y método seguido

Tras explorar las opciones disponibles, se ha elegido utilizar el motor Unity⁶. Se trata de una estupenda y popular herramienta de desarrollo de videojuegos que cuenta con más de 25 plataformas de exportación, donde se incluye Android. Además, el motor ofrece un plan personal gratuito usando su versión más actualizada (aunque con algunas reglas adicionales cuando el videojuego supera cierta cantidad de ingresos). Con Unity se han desarrollado juegos tan colosales como Hearthstone⁷ (inicialmente en PC, que más tarde dio el salto a dispositivos móviles), Pokémon GO⁸ o Monument Valley 2⁹. Por supuesto, una razón de peso a la hora de elegir utilizar Unity es haber cursado la asignatura de Introducción a videojuegos en dispositivos móviles, donde se ha utilizado este motor.

Como ya hemos dicho previamente, el desarrollo de este videojuego centra en su salida en Android, pero gracias a este programa, se podrá exportar a otras plataformas, como iOS, WebGL, etc. [\[4\]](#)

Respecto a la metodología utilizada para el desarrollo de la aplicación, se estudian las opciones más comunes: Modelo Waterfall (en cascada), desarrollo rápido de aplicaciones o desarrollo ágil. El primero se basa en una serie de fases que se cierran de forma secuencial y donde no hay retroalimentación entre ellas. Un proceso en cascada estará fijado por fechas límites y presupuestos. [\[5\]](#)

Se descarta el Modelo Waterfall de entrada, porque sólo es aplicable cuando están totalmente cerrados los requisitos y no van cambiar, y este no es el caso, ya que a medida que se desarrolle el juego, podrán ir cambiando en función de las dificultades que nos encontremos.

El desarrollo rápido de aplicaciones se centra en la obtención de un prototipo funcional de una aplicación para posteriormente ir mejorándolo incluyendo más

⁶ Unity es actualmente la plataforma líder del mundo para crear y operar contenido interactivo 3D en tiempo real (RT3D). El 71% de los 1.000 mejores juegos en dispositivos móviles de 2020 fueron creado con Unity. [Página oficial](#).

⁷ Blizzard entertainment (Videojuego). El juego de cartas digital con más éxito actualmente. [Página oficial](#).

⁸ Niantic, Inc. (Videojuego). El juego del universo Pokémon que usa realidad virtual y mapas para su jugabilidad. [Página oficial](#).

⁹ Ustwo Games (Videojuego). Un reclamado videojuego de puzzles para dispositivos móviles. [Página oficial](#).

funcionalidades y complejidad. Esto es una excelente opción en la que comenzar nuestro proyecto, mientras que el desarrollo ágil se caracteriza por utilizar una formación cíclica, donde en cada uno de los ciclos se obtenga un prototipo más completo y elaborado. En cada iteración del ciclo se incluye:

- Planificación
- Análisis de requerimientos
- Diseño, codificación
- Revisión y documentación.

Este método es muy adecuado para el desarrollo de aplicaciones móviles y será perfectamente viable utilizarlo después de obtener una primera versión funcional. Por tanto, y para resumir, comenzaremos utilizando el método de desarrollo rápido de aplicaciones, buscando obtener un primer modelo funcional de la aplicación, y posteriormente utilizar el método de desarrollo ágil para mejorarlo de manera cíclica. [6]

1.4. Planificación del Trabajo

Antes de realizar la planificación del trabajo, primero se deberán calcular los recursos necesarios para la elaboración del proyecto. En este tipo de trabajos existen tres tipos de recursos: software, hardware y humanos. Como único responsable de este proyecto, me encargaré de todos los perfiles abordables, por lo general son: jefe de proyecto, programador, analista/testeador, ilustrador y productor musical. Los dos últimos no se tendrán en cuenta, ya que se han adquirido los gráficos y sonidos mediante terceros.

Los recursos software son todos aquellos programas que se utilizarán para llevar a cabo el proyecto. Los recursos hardware son aquellos materiales físicos que se requieren para complementar el desarrollo. A continuación, se detallarán estos recursos en la siguiente tabla:

Recursos Software
• Unity: motor de desarrollo de videojuegos.
• Visual Studio Code: editor de código con herramientas integradas.
• Microsoft Word: editor de documentos de texto.

Tabla 1. Recursos Software.

Recursos Hardware
• Ordenador MSI GE63 Raider RGB 85F.
• Dispositivo Móvil One Plus 7 Pro.
• Otros dispositivos para testear la aplicación.

Tabla 2. Recursos Hardware.

Como recursos humanos, el proyecto tiene una duración estimada de 300 horas de trabajo, esto es, aproximadamente 3 horas de trabajo diarias. Para llevar un control de horas trabajadas, se propone hacer un seguimiento de tiempo de trabajo semanal,

de forma que se ejecutarán 21 horas cada 7 días, independientemente de los días laborables y festivos. Con toda probabilidad, el tiempo de trabajo se verá intensificado en la etapa de implementación.

Una vez detectados los recursos que se van a utilizar, se definirán las distintas etapas del proyecto y para la posterior clasificación y organización de las tareas a realizar para cada una de ellas, especificando además el tiempo estimado necesario para cada una de ellas. Este proyecto consta de 4 etapas bien diferenciadas:

1. Plan de trabajo: En esta etapa se definirá el concepto general del trabajo, lo que se quiere conseguir y los objetivos para cumplirlo, lo que necesitaremos para ello y una planificación para llevarlo a cabo.
2. Diseño: Que consiste en los primeros pasos para darle forma al desarrollo que se quiere realizar. Empezando por un estudio del público objetivo, la descripción funcional del producto, planteamiento de las pantallas y diseño arquitectónico.
3. Implementación: Una vez considerado el diseño del producto, se comienza el propio desarrollo del mismo. Empezando por un diseño funcional básico, como mencionábamos, al que posteriormente se le añadirán recursos gráficos y sonoros. Después se implementarán dos versiones en las que se ejecutará retroalimentación a la par que se añadirán funciones adicionales. Por último, se lanzará el producto en los dispositivos móviles para realizar numerosas pruebas.
4. Entrega final: A lo largo de todo el TFM se irá recopilando toda la información en un documento, se revisarán los productos finales y se prepararán todos los entregables.

La etapa que puede dar lugar a complicaciones en la planificación es la de implementación, ya que es en la generación de código donde pueden aparecer errores y dificultades en la generación de nuevas funcionalidades. Para solventarlo, se ha creído conveniente asignar un mayor porcentaje de horas de trabajo estimadas en esta etapa, de forma que haya un margen de maniobrabilidad frente a los obstáculos que puedan suceder.

En la siguiente página, se presenta el diagrama de Gantt de la planificación de este TFM, donde podemos ver apreciar las tareas distribuidas en el tiempo, junto con la cantidad de horas de trabajo estimadas para su ejecución.

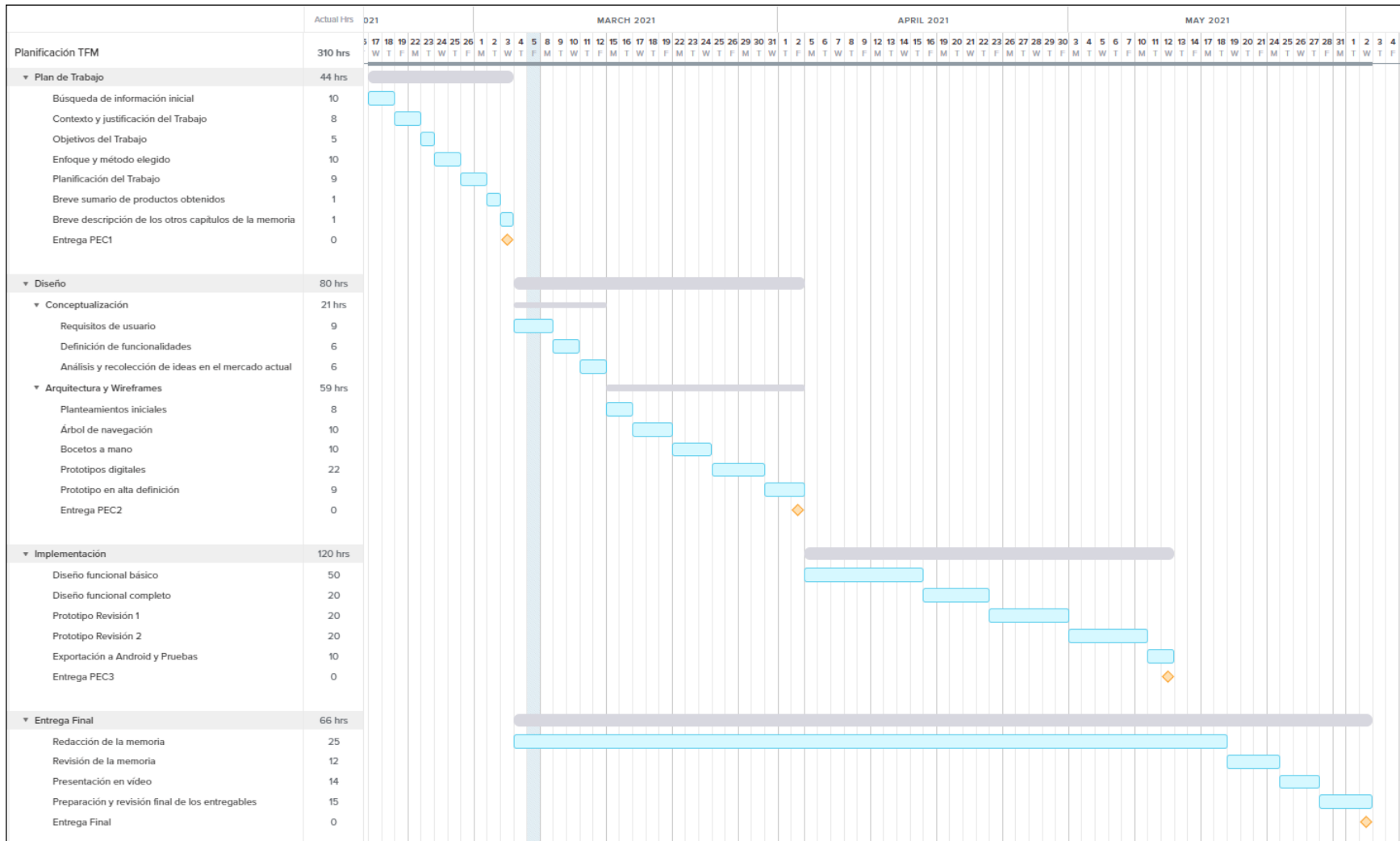


Ilustración 1. Diagrama de Gantt del TFM

1.5. Breve resumen de productos obtenidos

Al seguir la planificación representada en la página anterior con éxito conseguiremos un videojuego, inicialmente pensado para Android, que más tarde se podrá mejorar, comercializar y expandir a otras plataformas.

Al acabar este proyecto los entregables con los que contaremos son:

- Documento donde se registre con detalle todo el proceso del desarrollo, desde su planteamiento inicial hasta su implementación y testeo.
- El videojuego de tipo idle o inactivo. Estará pensado para disfrutarlo en dispositivos Android, por tanto, se entregará un archivo con extensión apk.
- Un video de presentación del proyecto explicando, a grandes rasgos, el transcurso en el desarrollo del proyecto, así como una pequeña muestra del producto.

1.6. Breve descripción de los otros capítulos de la memoria

En principio, los capítulos de este TFM siguen la planificación estudiada anteriormente. A continuación, se alistarán un índice orientativo del resto de capítulos que conforman el proyecto.

2. Diseño
 - 2.1. Conceptualización
 - 2.1.1. Requisitos de usuario
 - 2.1.2. Definición de funcionalidades
 - 2.1.3. Análisis y recolección de datos en el mercado actual
 - 2.2. Arquitectura y Wireframes
 - 2.2.1.1. Planteamientos iniciales
 - 2.2.1.2. Árbol de navegación
 - 2.2.1.3. Bocetos a mano
 - 2.2.1.4. Prototipos digitales
 - 2.2.1.5. Prototipo en alta definición
3. Implementación
 - 3.1. Diseño funcional básico
 - 3.2. Diseño funcional completo
 - 3.3. Prototipos
 - 3.3.1. Revisión 1
 - 3.3.2. Revisión 2
 - 3.4. Exportación a Android y Pruebas

Esta clasificación de capítulos no es definitiva, y puede cambiar a lo largo del desarrollo del proyecto. Con probabilidad, se añadirán capítulos para proporcionar toda la información necesaria relativa al producto. Además, este apartado no aparecerá en la memoria completa del proyecto, ya que no será relevante una vez tengamos el resto de capítulos en él.

También es adecuado comentar que el contenido escrito en esta memoria de plan de trabajo podrá ser modificado en versiones posteriores a la misma, siempre con el propósito de mejorarla.

2. Diseño.

En este capítulo se realizará un estudio de los factores relativos al diseño del producto. En primer lugar, comenzaremos conceptualizando el producto, donde investigaremos cuál será el usuario potencial y llevaremos a cabo un estudio del mercado actual que nos servirá de ayuda para definir las funcionalidades básicas que debe tener nuestro producto. Por otro lado, se estudiará la estructura principal que tendrán las pantallas de nuestro programa y se elaboraremos numerosos prototipos, desde bocetos a mano hasta prototipos de alta fidelidad.

2.1. Conceptualización

El primer paso para elaborar el diseño de un proyecto es conceptualizarlo, es decir, el paso de una idea subjetiva a una materialización de la misma teniendo el prototipo y otras herramientas como medio para plasmarla. El ensayo y error en la conceptualización resulta de lo más normal y lógico.

2.1.1. Investigación y requisitos de usuario

En este apartado definiremos el público objetivo que suponemos usará la aplicación. En primer lugar, presentaremos dos arquetipos de personas que puedan estar interesadas en este juego, ejemplificándolos y aportando datos sobre ellos afines a la realidad. Una vez queden definidas dichas personalidades, quedará retratado un escenario por cada arquetipo en el que se relevan posibles motivos para acceder a la app y usarla la mayor cantidad de tiempo posible.

En general, este tipo de juegos es utilizado por personas entre 16 y 30 años, donde hay una mayoría de usuarios con género masculino, este será entonces nuestro público objetivo.¹⁰

En el estudio del público objetivo de nuestra aplicación podemos encontrar dos perfiles bien diferenciados: por un lado, usuarios que tengan experiencia en juegos con mucho contenido y les guste explorar las posibilidades que les brinde la app; y por otro, usuarios que no están especializados en mecánicas complejas y que busquen un entretenimiento de forma casual, con algo que enganche y no les haga pensar demasiado.

A continuación, se presentarán las fichas de persona de estos perfiles.

¹⁰ De manera subjetiva, basado en mi experiencia de usuario.

Carlos Álvarez



Edad: 28
Profesión: Informático
Estado: Soltero
Población: Madrid

"Busco un juego para el móvil en el que pueda desarrollarme para cuando no estoy en casa y en mis ratos libres"

Hardware

- Portátil MSI GE63
- Dispositivo Móvil Samsung S10+
- Consola Playstation 4

Situación actual

Carlos se ha mudado recientemente a un piso de alquiler con su novia. Se encuentra trabajando de informático en una empresa cercana a su nueva casa, con horario de oficina. Cobra 1400 euros al mes. No tiene carnet de conducir, ya que no lo considera necesario, generalmente se transporta utilizando el metro. Por la tarde, se dedica a ir al gimnasio, hacer planes con su novia o con sus amigos, tareas del hogar, y en su tiempo libre, jugar a videojuegos.

Sobre videojuegos

Acostumbrado a utilizar juegos de gran profundidad estratégica en consolas y ordenador. En su dispositivo móvil, se encuentra buscando algún juego con que llenar su tiempo libre. Ha probado varios pero no le convencen o bien porque no engancha lo suficiente o porque no tiene la suficiente profundidad a la que él está habituado.

Aplicaciones habituales

- WhatsApp
- Spotify
- Wild Rift
- Genshin Impact
- Brawl Stars
- Tap Titans 2.

Estilo de vida

Poco tiempo libre Mucho tiempo libre

Vida en casa Vida en la calle

Ahorrador Derrochador

Vida Activa Vida pasiva

Gustos

Hacer Deporte

Redes sociales

Ver Televisión

Ocio en sociedad

Eventos culturales

Videojuegos

f t in p y i

Ilustración 2. Ficha de persona: Jugador experimentado.¹¹

Alberto García



Edad: 19
Profesión: Estudiante de Ingeniería Industrial.
Estado: Soltero
Población: Almería

"Busco un juego que me entretenga y enganche para dedicarle ratos cortos"

Hardware

- Ordenador de sobremesa (a compartir con la familia).
- Xiaomi Mi 8
- Nintendo Switch

Situación Actual

Alberto vive con sus padres y sus tres hermanos. Él tiene una vida muy ajetreada, su tiempo lo ocupa mayormente en las clases; cursos de inglés, deporte y música; y además pertenece a una familia numerosa. Por la tarde suele hacer planes con sus amigos, estudiar en la biblioteca o jugar a videojuegos. También se encuentra sacándose el carnet de conducir, actualmente usa el autobús para transportarse.

Sobre Videojuegos

Alberto acostumbra a utilizar los videojuegos como medio de entretenimiento por encima de todo, le gustan los juegos competitivos. Él está buscando una aplicación que no le exija mucho tiempo de uso, pero que sea atractivo, vistoso y no le haga pensar demasiado.

Aplicaciones habituales

- Telegram
- Brawl Stars
- Spotify
- Call of Duty: Mobile
- Wallapop
- Tik Tok

Estilo de vida

Poco tiempo libre Mucho tiempo libre

Vida en casa Vida en la calle

Ahorrador Derrochador

Vida activa Vida pasiva

Gustos

Hacer deporte

Redes sociales

Ver Televisión

Ocio en sociedad

Eventos culturales

Videojuegos

f t in p y i

Ilustración 3. Ficha de persona: Jugador casual.¹¹

¹¹ Fichas de persona creada utilizando la versión de prueba de <https://xtensio.com/>
Las fotografías pertenecen a personas irreales generadas digitalmente, obtenidas de <https://thispersondoesnotexist.com/>

Una vez mostradas las fichas de persona por perfil, se procederá a retratar un escenario a cada uno de estos arquetipos. De esta forma, se podrán revelar posibles motivos para acceder a la aplicación, probarla y dedicarle tiempo de uso en el tiempo.

1. Escenario de Carlos: Jugador experimentado.

Carlos lleva tiempo buscando un juego que satisfaga su necesidad de llenar esos tiempos libres, y que a la vez le aporte esa sensación de progresión y de desarrollo del personaje habituales en juegos de rol. Él ha probado numerosos juegos que han acabado por no funcionar, o bien por resultar un juego de tipo inactivo demasiado sencillo y con poca o nulas opciones de elección en la que realizar la progresión, o bien porque resultaba demasiado inmersivo en el sentido de requerir mucha concentración y atención constante (para eso ya juega desde el ordenador o la consola).

Tras una exhaustiva búsqueda en la tienda de aplicaciones encontró nuestro juego. Al abrir y probar la aplicación, Carlos se ha encontrado un juego Idle con una ambientación que le llama la atención y un conjunto de opciones de gran variedad con el que podrá realizar diferentes combinaciones. Además, surge ese efecto adictivo gracias a la progresión constante a lo largo del tiempo, viendo como mejora debido a sus elecciones y a los beneficios obtenidos al volver a abrir la aplicación (recompensas después de un periodo con la aplicación cerrada).

A medida que su progresión avanza, consigue desbloquear nuevos contenidos que completan el desarrollo de su personaje y puede ver como mejora su puntuación respecto a otros usuarios, gracias a una tabla de clasificación general. Esto hace que aumente en gran medida el interés en seguir utilizando la aplicación.

2. Escenario de Alberto: Jugador casual.

La vida de Alberto es muy caótica y ajetreada, no cuenta con mucho tiempo libre. Sin embargo, debido a que no le gustan los tiempos vacíos, está buscando un juego para el dispositivo móvil para entretenerse de forma ocasional, además, le gustaría sentir que existe algún tipo de progresión, de lo contrario se tornaría repetitivo, que es su problema actual.

Navegando por la tienda de aplicaciones, encontró el juego por casualidad y se dispuso a darle una oportunidad. En un primer vistazo, encontró un juego con un diseño vistoso y limpio que le llamó la atención rápidamente.

El juego resulta sencillo al principio, y en él puede seguir una progresión constante, incluso aunque no esté jugando. A lo largo del tiempo aparecen nuevas opciones, pero las misiones de seguimiento y el contenido explicativo del juego hacen que no exija pensar demasiado (ya que su mayor objetivo es entretenerse sin grandes complicaciones).

2.1.2. Definición de funcionalidades

A raíz del análisis desarrollado en el apartado anterior, podremos definir las funcionalidades principales de nuestra aplicación, tratando de encontrar los puntos fuertes que buscan los dos perfiles estudiados.

Para el primer perfil, las recomendaciones para captar y mantener a un usuario será realizar una progresión que consiga ese efecto adictivo característico de los juegos

inactivos y utilizar mecánicas frecuentes en juegos de rol en los que el usuario pueda tomar sus propias decisiones. En cambio, para el segundo, las recomendaciones para captar y mantener a un usuario será realizar una aplicación con una buena presentación, que cuente con una interfaz de usuario amigable y que no sea necesario dedicarle grandes cantidades de tiempo.

Después de realizar un análisis sobre el público objetivo de la aplicación, encontrando dos tipos de perfiles y sus necesidades, podemos considerar las siguientes funcionalidades básicas:

- **Front-end:**

- **Una interfaz de usuario clara, limpia y atractiva.** El juego se basa enteramente en el uso de una interfaz de usuario compleja, por eso, será necesario que se encuentre bien organizada y estructurada.
- **Gráficos y efectos llamativos y que guarden una temática de fantasía.** La primera impresión influye muchísimo en el usuario para que decida seguir utilizando o no la aplicación, por tanto, el juego deberá ser atractivo a los ojos del jugador (se trata de nuestra carta de presentación).
- **Sistema de seguimiento, tutorial y ayuda en el juego.** Al tratarse de un juego en un dispositivo móvil, un usuario no poseerá demasiada concentración a la aplicación (hoy en día, los usuarios acostumbran a utilizar el móvil mientras realizan otras acciones, y también de cambiar de aplicación rápidamente). Los jugadores no experimentados se sentirán mucho más cómodos al recibir instrucciones a medida que avanza su progresión.

- **Back-end:**

- **Guardado de datos del personaje.** Para que el usuario pueda conseguir una progresión correctamente, será necesario que la aplicación guarde continuamente los datos del personaje. Como funcionalidad avanzada, se podría implementar un sistema de login en la aplicación, de forma que la información del personaje de un usuario esté vinculada a su cuenta.
- **Contar con un sistema 'idle',** que funcione y donde se generen beneficios en el tiempo, aunque el usuario no esté utilizando la aplicación. Se trataría de que cuando el usuario abra la aplicación se encuentre recompensas según el tiempo que se ha encontrado 'inactivo'. Estas deberán ser lógicas y congruentes para una correcta progresión del juego.
- **Añadir mecánicas de progresión** característicos de juegos de rol. Será importante añadir diferentes elementos para añadir un componente diferencial respecto al mercado de juegos 'idle'. Estos aportarán profundidad y brindarán al usuario la satisfacción de tomar decisiones en su desarrollo del personaje.

2.1.3. Análisis y recolección de datos en el mercado actual

Una vez definidas las funcionalidades básicas que debe tener nuestra aplicación, se procede a realizar una búsqueda en el mercado actual para encontrar ideas exitosas para estas funcionalidades, o encontrar nuevas funcionalidades que aporten valor al proyecto. En este caso, se contemplarán tres aplicaciones que utilicen 3 de nuestros puntos básicos por separado.

En primer lugar, un videojuego de tipo inactivo puro: Cookie Clickers™, que es el precursor del propio género. En segundo lugar, un juego de rol con diferentes mecánicas de personalización del personaje: Eternium. Y, en tercer lugar, un juego de tipo inactivo que también guarda diferentes mecánicas en el desarrollo del personaje: Tap Titans 2; este juego será el que guarde mayor similitud al nuestro.

1. Cookie Clickers™

Este es uno de los primeros y de los más famosos juegos de estilo inactivo que se crearon, en su sencillez reside su grandeza. Una mecánica muy simple y unos detalles coloristas y llamativos han hecho que la aplicación genere millones de descargas.

Su pantalla principal es simplemente una galleta gigante, preparada para recibir la mayor cantidad de clics posibles, información de las galletas obtenidas y una media de las que se consiguen por segundo, abajo tienes una serie de opciones: las más relevantes son:

- La tienda donde comprar mejoras para aumentar el ritmo de generación de galletas por segundo.
- Clasificación de galletas totales obtenidas, global y de los amigos de Facebook que también usen la aplicación (requiere loguearse en Facebook).
- Otras ventajas de pago para obtener mayor cantidad de galletas, esta será la forma de monetizar de la aplicación.



Ilustración 4. Pantalla principal de Cookie Clickers™

Con la observación de este juego podemos sacar en claro que es importante disponer de un espacio donde pulsar la pantalla repetidamente (en nuestro caso, para realizar ataques). Sería conveniente que nuestra pantalla principal esté despejada y con una presentación que luzca vistosa.

También será buena idea incitar a la competitividad de los usuarios brindando una clasificación global, incitando así a jugar más por el afán a escalar puestos respecto a los demás. También provocará placer el usuario el hecho de ver como los ingresos aumentan exponencialmente.

2. Eternium

Eternium es un videojuego para dispositivos móviles de tipo acción y rpg. En él controlarás a un personaje a lo largo de una historia, donde avanzarás por diferentes

niveles y tendrás que enfrentarte a enemigos cada vez más poderosos. Para ello, el juego cuenta con varias herramientas de desarrollo del personaje, que es el factor a analizar.

En primer lugar, al elegir un personaje, deberás elegir también una de las tres clases disponibles, que contarán con opciones de personalización diferentes.



Ilustración 5. Pantalla de equipamiento de Eternium.

Las diferentes opciones para mejorar al personaje son:

- Un sistema muy completo de equipamiento. El personaje cuenta con 14 huecos para equiparse objetos. Cada pieza de equipo puede tener varias calidades y beneficios diferentes. Además, también existen varias formas para mejorar y modificar el equipamiento, aportando flexibilidad en nuestras elecciones.
- El poder del personaje está basado en tres atributos esenciales: daño, resistencia y recuperación.
- Un sistema de habilidades, activas y pasivas. El personaje cuenta con 8 huecos de habilidades a elegir dentro de un gran panel de habilidades (diferentes en cada clase). Además, las habilidades también pueden ser mejoradas.

Este sistema podría ser tomada en nuestro videojuego como mecánicas para desarrollar al personaje. Un sistema de equipamiento, donde el jugador pueda elegir sus propias piezas según le vayan cayendo. También es totalmente plausible la inclusión de habilidades, tanto activas como pasivas en la que el personaje pueda realizar sus elecciones de uso.

3. Tap Titans 2

Uno de las aplicaciones principales que podemos tener en cuenta, y con el éxito que tiene es Tap Titans 2. Se trata de un videojuego para dispositivos móviles que funciona en pantalla vertical.

El sistema de progresión del juego se basa en ir eliminando enemigos, que nos otorgarán dinero con el que poder mejorar a nuestro personaje y contratar otros héroes que nos ayudarán en combate. También existen los artefactos, que se pondrán usar para conseguir beneficios permanentes. El juego también cuenta con la posibilidad de resetear y empezar a nivel 1, aportando ventajas respecto a la partida anterior. Además, nuestro personaje también podrá usar diferentes habilidades que nos ayudarán en la progresión.

Su pantalla principal presenta en primera plana el enemigo contra el que nos enfrentamos, y nuestro personaje, de espaldas a la pantalla, enfrentados a él, también aparecen los héroes que hayamos reclutado. El resto de información es de interfaz de usuario, por orden de arriba abajo, de izquierda a derecha:

- Zona superior: Acceso a pantalla de opciones; Acceso a la pantalla de Clan; Acceso a la pantalla de Clan; Nivel del enemigo; Acceso a la pantalla de torneo; Vida del enemigo
- Zona Central e Inferior: Información sobre nuestra progresión: daño de ataque, daño por segundo, maná, etc.; Habilidades activas: aportan beneficios puntuales, hay que esperar un cierto tiempo para poder volver a activar las habilidades; Acceso a diferentes opciones para el desarrollo de nuestro personaje.



Ilustración 6. Pantalla principal de Tap Titans 2.

Tap Titans 2 es un gran ejemplo de una buena estructuración en los datos a mostrar. Manteniendo una pantalla principal bien despejada para pulsar rápidamente, como en Cookie Clicker, y a su vez, permite acceder a un gran número de opciones utilizando pestañas desplegadas que además pueden maximizarse para la comodidad del usuario.

Este juego se tendrá en cuenta a la hora de diseñar una interfaz de usuario eficiente para nuestro juego.

2.2. Arquitectura y Wireframes

2.2.1. Planteamientos iniciales

Una vez definidas las funcionalidades básicas del proyecto y habiendo analizado otras aplicaciones similares a la nuestra, comenzaremos detallando que características nos interesará tener en nuestras pantallas:

- Pantalla principal: Convendrá disponer de una pantalla principal limpia, con la información indispensable y acceso al resto de opciones bien estructuradas. Además, la vista de esta pantalla tiene que ser atractiva para el jugador, ya que será donde pase más tiempo.

En el centro se ubicarán nuestros enemigos, quizás podría ser una buena opción ubicar a más de un enemigo. Por ejemplo, que puedan aparecer de uno a tres enemigos de manera aleatoria al pasar a la siguiente zona.

La información básica a mostrar podrá ser: vida, maná, nivel y experiencia del jugador; nivel de la zona en el que se encuentra el jugador (por lo tanto, nivel de los enemigos); oro almacenado; daño infligido al pulsar, daño automático por segundo.

Las opciones e información disponibles estarán en el lado superior o inferior de la pantalla. Por ejemplo, acceso a mecánicas de progresión y habilidades en el lado

inferior e información relativa al nivel del enemigo, vida y mana del personaje en el lado superior.

- Mecánicas de progresión: Como mecánicas de progresión, elegimos en una primera instancia características muy recurridas en juegos de rol.
 - Atributos: Un sistema de atributos simple, donde cada elemento quede bien diferenciado y guarde variedad. Al elegir la opción de atributos, se desplegará una ventana con la lista de atributos, junto con el icono correspondiente para que el jugador pueda gastar sus puntos de atributos. También se tendrá que mostrar en pantalla que le aporta al personaje cada uno de ellos.
 - Habilidades: Un completo conjunto de habilidades a elegir. En este juego, no se elegirá una clase inicial, si no que cada usuario podrá elegir su propio camino, todas las opciones se podrán ir desbloqueando, siempre que se cumplan los requisitos (por ejemplo, desbloquear una habilidad si has aprendido otra habilidad anterior). Una buena idea para mostrar las habilidades será representarlas con un estilo de árbol genealógico.

Las habilidades irán de la mano con los atributos, habrá habilidades tanto activas, o activables cada cierto tiempo, como pasivas, o beneficios permanentes.

- Equipamiento: Se realizará un sistema de equipamiento. El personaje irá ganando botines con el que conseguirá mejorar su personaje y aumentar su ritmo de progresión. Como en muchos juegos de rol, la generación de nuevos equipamientos será aleatoria, en función del nivel y del tipo de enemigo derrotado, podrán caer objetos más simples o más poderosos. Para presentarlo se tendrá que, por un lado, mostrar lo que el personaje tiene equipado actualmente, y que pueda verificar que le aporta cada una de sus piezas equipadas. Por otro lado, tendrá que poder ver el inventario de los botines que ha conseguido, y además, tener la opción de equiparse o deshacerse de cada pieza.
- Opciones: El juego deberá contar con un menú de acceso a las opciones, donde controlar algunos aspectos como el sonido, información y ayuda acerca del juego, o cualquier aspecto no relacionado con la jugabilidad de la aplicación.
- Un sistema de reseteo: Aunque aún no está definido, sería conveniente que el juego contara también con un sistema de reseteo, típico de los juegos inactivos. Esto es, volver a empezar el juego, pero empezando con ciertas ventajas, que serán mayores entre que más se haya avanzado en la partida jugada.
- Clasificación global: Aunque no es estrictamente necesario, será muy interesante agregar al juego una pantalla de clasificación en el que los usuarios puedan apreciar su posición en comparación con el resto de usuarios que

Estas mecánicas serán accesibles desde botones en la sección inferior de la pantalla principal, al pulsar un botón, se desplegará una ventana de interfaz gráfica en la que aparezcan las opciones pertinentes.

- Misiones de tutorial y logros: Para acabar y de manera opcional, se le puede aportar un valor añadido al juego agregándole misiones que inciten al jugador a conseguir las para obtener pequeños beneficios extra y, además, un sistema de logros que completen la experiencia de juego. Sin embargo, para obtener este sistema, será importante recopilar una gran información que el usuario realice sobre el juego para

podemos hacer el seguimiento de estas (por ejemplo, número de pulsaciones, cantidad de enemigos derrotados, número de reinicios realizados, etc.)

2.2.2. Árbol de navegación

En este apartado se plasmarán las secuencias de pantallas necesarias para que la aplicación pueda funcionar correctamente y eficazmente. Tal y como se desarrolla en el apartado anterior, y además teniendo en cuenta los puntos tratados en la etapa de conceptualización, el árbol de navegación que se plantea se presenta a continuación:

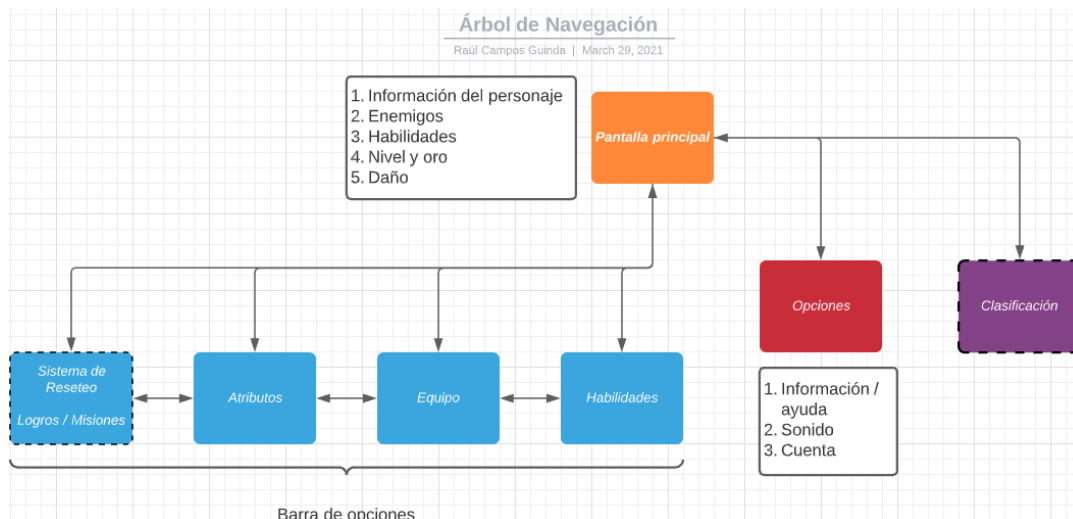


Ilustración 7. Árbol de navegación.¹²

En él nos encontramos la pantalla principal, el bloque de pantallas de desarrollo de personaje, en azul, que podrán cambiarse entre ellas gracias a la barra de accesos (o toolbar) que se encontrará en la zona inferior. Y, por otro lado, tendremos por separado las pantallas de opciones y de clasificación. Todas las pantallas secundarias podrán cerrarse o minimizarse para dar siempre paso a la pantalla principal.

Notar que dos de los bloques están bordeados con líneas discontinuas, expresando así, que no serán un requisito fundamental en nuestro proyecto, pero que podrán agregar un gran valor al contenido del juego.

2.2.3. Bocetos a mano

Ahora que se ha decidido como se relacionarán las pantallas de nuestra aplicación, tendremos que empezar a dar forma a cada una de las pantallas. Para ello, se aconseja coger papel y lápiz, dar rienda suelta a nuestra imaginación y proceder a realizar el llamado 'brain storming'. En los diseños iniciales, nuestro objetivo es organizar la información a mostrar de la forma más eficaz posible, la presentación queda en un segundo plano.

¹² Este árbol de navegación se ha diseñado utilizando la herramienta online <https://lucid.co/>

Aunque hoy en día existen un sinfín de herramientas para la creación de prototipos digitales de forma rápida, es importante empezar con bocetos a mano, ya que no existe ninguna limitación a la hora de diseñar nuestras pantallas, a excepción de la imaginación. Con herramientas digitales, en cambio, las utilidades que aporta cada programa son finitas, limitando las opciones del diseño. Una vez se tenga una idea aproximada de cómo se mostrarán, al menos, las pantallas más importantes de nuestra aplicación, se procederá realizar prototipos de diseños más completos utilizando herramientas de prototipado digital.

Comenzaremos realizando los primeros bocetos a mano, tratando de seguir las líneas (nunca mejor dicho) mencionadas en los apartados anteriores (aunque estos bocetos han sido realizados en paralelo a la escritura de esta memoria).

Después de varios intentos de dibujar y borrar, se muestran los últimos bocetos realizados. Por orden de aparición, se encuentran la pantalla principal, junto con un diseño alternativo, la pantalla de atributos, la pantalla de equipo y la de habilidades.

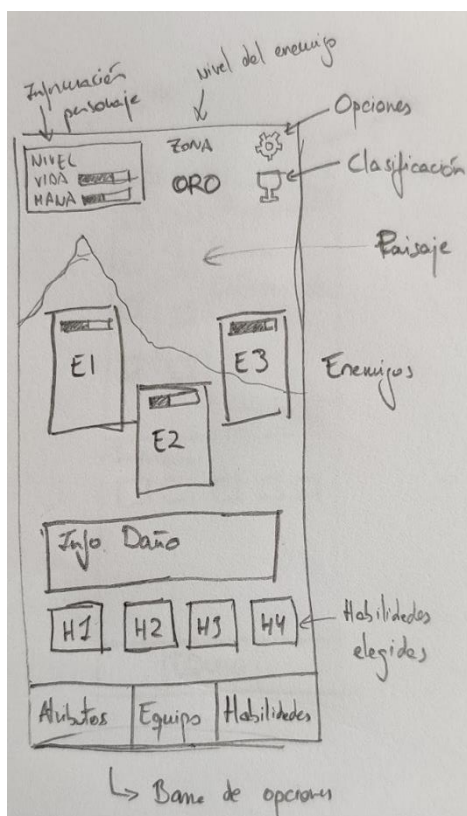


Ilustración 9. Boceto a mano de la pantalla principal.

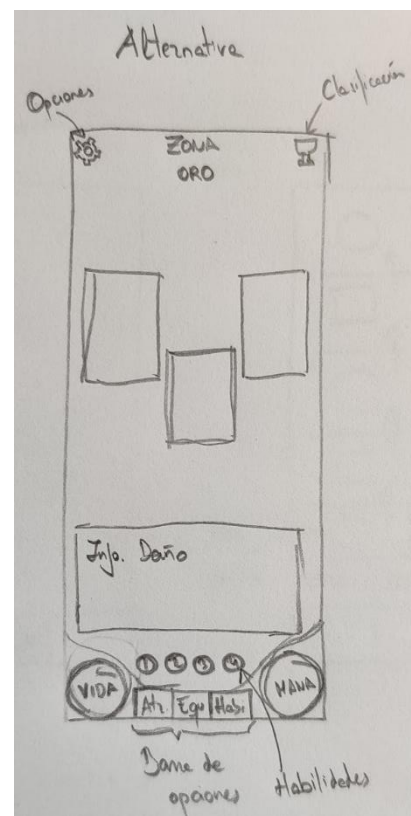


Ilustración 8. Boceto a mano de la pantalla principal: diseño alternativo.

A la hora de dibujar el boceto de la pantalla principal, se ha procurado tener en cuenta los detalles mencionados en capítulos anteriores, de forma que la zona donde pulsar para atacar sea lo más grande posible y se muestre la información y las opciones indispensables. Además, en el boceto alternativo se ha tratado de despejar aún más la zona de pulsaciones, empujando el resto de opciones visibles.

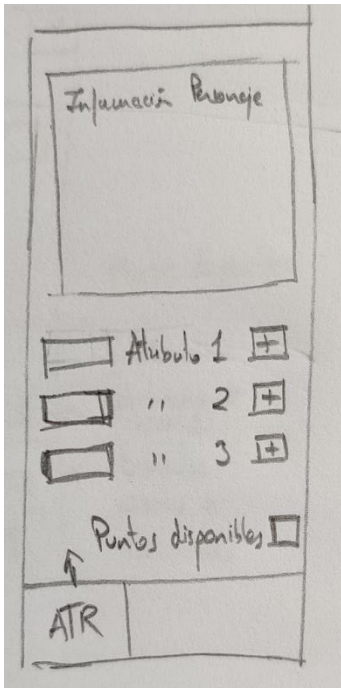


Ilustración 10. Boceto a mano de la pantalla de atributos.

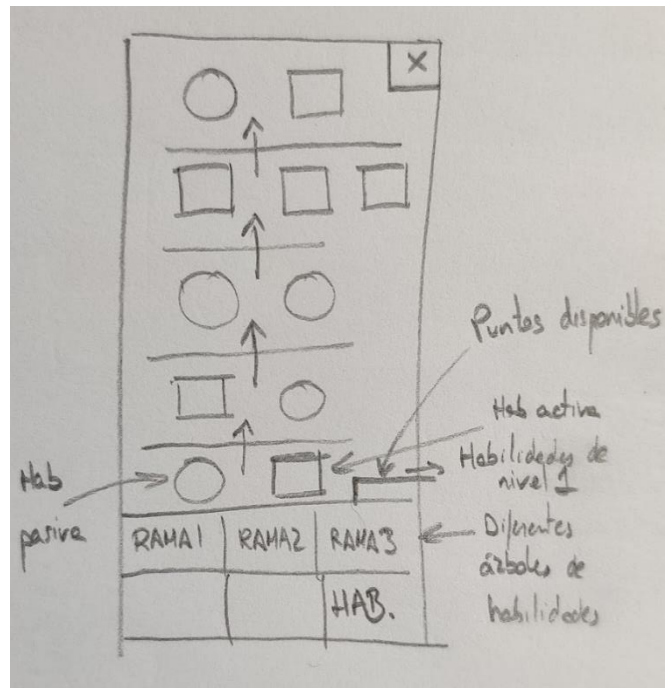


Ilustración 11. Boceto a mano de la pantalla de habilidades.

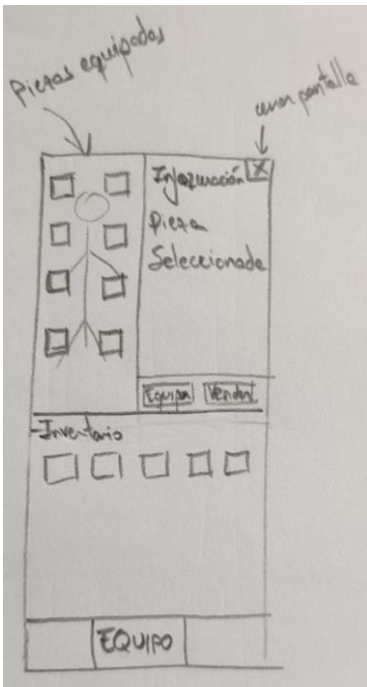


Ilustración 10. Boceto a mano de la pantalla de equipo.

Los bocetos del resto de pantallas se han planteado siendo fieles a como se describieron en capítulos anteriores, sin excederse en detalles. Los bocetos de las pantallas de opciones y de clasificación no se han considerado necesarias, ya que serán muy sencillas y no requieren de un estudio previo.

Las diferentes secciones que aparecen en las pantallas se explicarán en más detalle en el siguiente capítulo.

2.2.4. Prototipos digitales

En este apartado se presentarán, para las pantallas principales, unos prototipos más desarrollados que los bocetos iniciales. Para la realización de los prototipos, se han probado varias herramientas online de diseño de prototipos, pero finalmente se ha escogido NinjaMock por la flexibilidad y eficacia que presenta para hacer los diseños.

En NinjaMock, puedes vincular diferentes proyectos al crearse una cuenta online en esta página. A la hora de crear un proyecto, podrás establecer un conjunto de pantallas a las que se le van añadiendo elementos de pantalla (botones, etiquetas, espacio de fotos, etc.). Estos elementos, a su vez, pueden fácilmente modificarse en varios aspectos, como apariencia, diseño, fuente, vínculos, etc.

Para cada pantalla, analizaremos brevemente cada elemento visual utilizando números de referencia:

- Pantalla principal

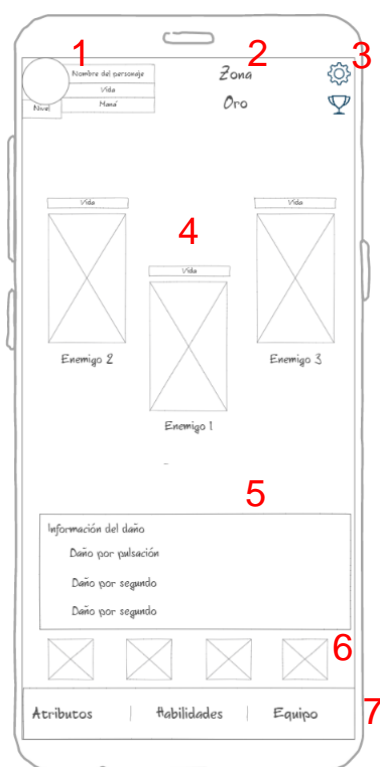


Ilustración 11. Prototipo digital de la pantalla principal.

Describiendo los elementos que aparecen en esta pantalla en orden de arriba abajo y de izquierda a derecha nos podemos encontrar: Sección de información del personaje (1), donde se ubicará el nombre, nivel, vida y maná del personaje, opcionalmente podría incluirse un avatar; El nivel de la zona, que indicará el nivel de los enemigos actuales, y justo debajo la cantidad de oro coleccionada por el jugador (2); el acceso a las pantallas de opciones y de clasificación (3); los enemigos estarán ubicados en la zona central de la pantalla, ya que será el mecanismo principal del propio juego, se define el espacio para hasta tres enemigos en pantalla, junto a sus barras de vida actuales (4); en una sección inferior se mostrará información básica del daño efectivo del jugador (5); la selección de habilidades a utilizar, el usuario podrá elegir un total de 4 habilidades que ya tenga aprendidas para usar en combate (6); por último, uno de los elementos más

importantes, se trata de un toolbar con el que acceder a las diferentes mecánicas de desarrollo del personaje: atributos, habilidades y equipo (7).

- Pantalla de atributos.

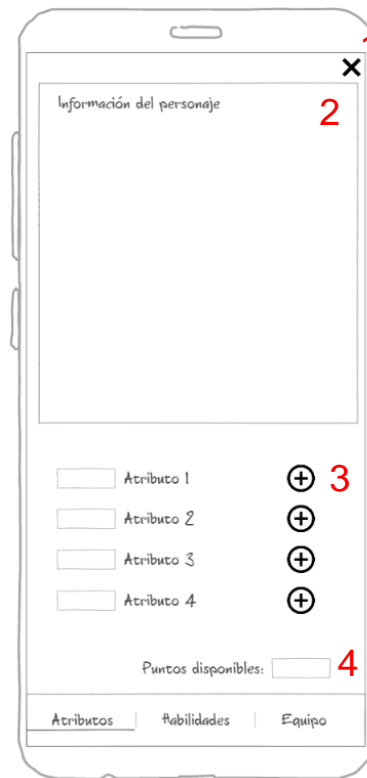


Ilustración 12. Prototipo digital de la pantalla de atributos.

Esta pantalla tiene dos propósitos para el jugador, proporcionar una información detallada del personaje (2) y la ubicación de los puntos disponibles (4) a repartir entre los atributos existentes (3). Para ello, se mostrará un botón "+" al lado de cada atributo mientras quede algún punto disponible, si no los hubiera no deben aparecer. La pantalla, además, cuenta con un botón "X" (1) para volver a la pantalla principal, o si el usuario lo prefiere, acudir a la pantalla de habilidades o de equipo.

- Pantalla de habilidades

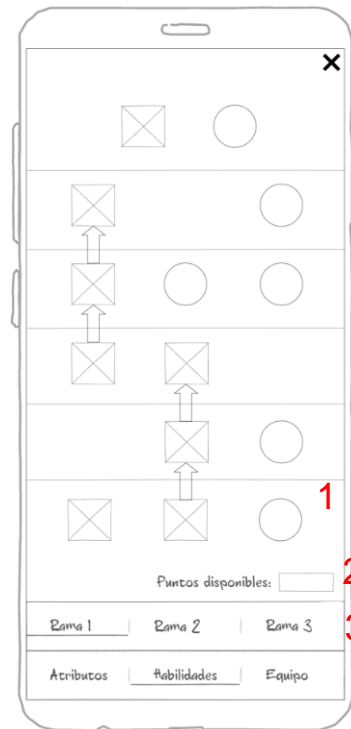


Ilustración 13. Prototipo digital de la pantalla de habilidades.

En esta pantalla nos encontramos con el mismo icono “X” para volver a la pantalla principal y el toolbar para navegar entre las pantallas de desarrollo del personaje. Por otro lado, los elementos que aparecen en esta pantalla son: listado de todas las habilidades disponibles para aprender, estructuradas por filas, de forma que no se puedan aprender habilidades de una fila superior hasta que se gasten X puntos disponibles (2) sobre habilidades de filas inferiores (1). Además, se encuentra un segundo toolbar donde se podrán seleccionar ramas diferentes e independientes (3), esto dará una mayor libertad en la personalización del personaje.

Esta pantalla está inspirada en los diseños antiguos que aparecían en el videojuego World of Warcraft, en la ventana de talentos, y que también es muy recurrido en otros juegos del mismo tipo:



Ilustración 14. Antigua ventana de talentos en World of Warcraft.

- Pantalla de equipo:

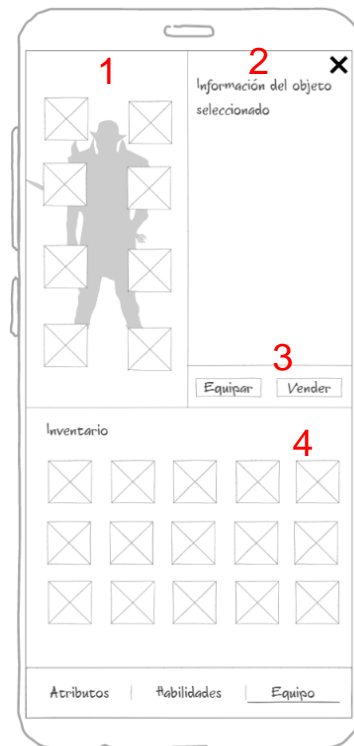


Ilustración 15. Prototipo digital de la pantalla de equipo.

Esta pantalla se encuentra dividida en 3 secciones principales: en primer lugar, el espacio donde se ubican las piezas actualmente equipadas por el personaje (1); en segundo lugar, un cuadro de información donde se muestran las características sobre el objeto seleccionado (2), tanto piezas situadas sobre el personaje como del inventario, en esta sección también se podrá decidir si la pieza se equipa sobre el personaje o se vende por una cierta cantidad de oro (3); por último, el espacio donde se situarán los objetos que se vayan recogiendo a lo largo de la aventura (4). La pantalla también cuenta con el botón “X” y el toolbar, al igual que sus pantallas ‘hermanas’.

2.2.5. Prototipo en alta definición

Para aportar una idea más cercana a la realidad del aspecto que podría tener el juego, se aporta, a continuación, un prototipo en mayor definición de la pantalla principal:



Ilustración 16. Prototipo en alta definición de la pantalla principal.¹³

Sin embargo, todavía no se cuenta con los recursos suficientes como para mostrar más información acerca del aspecto que tendrá finalmente nuestro videojuego.

Los prototipos aquí mostrados pueden no coincidir con el aspecto final del producto. A medida que se implemente el videojuego, puede sufrir cambios tanto en el front-end como en el back-end.

Los capítulos pertenecientes al diseño de este proyecto no son definitivos y podrán ser modificados y/o extendidos en versiones posteriores del documento.

¹³ Este prototipo en alta definición se ha creado utilizando el programa [Axure](#) en su versión de prueba gratuito

3. Implementación.

En este capítulo se documentará el desarrollo del producto estudiado en este proyecto. Esto es, la transformación de un diseño a un recurso, en este caso, un juego para dispositivos Android. En primer lugar, se dará a conocer el desarrollo que se ha llevado a cabo para conseguirlo, se mostrarán los recursos utilizados y seguidamente se mostrará la forma en la que se realizan las pruebas para comprobar el funcionamiento del producto.

3.1. Desarrollo

Como ya se mencionó anteriormente, se ha elegido el motor Unity para implementar el juego, ya que es una herramienta potentísima enfocada precisamente a lo que se pretende realizar. Por tanto, se comenzará creando un nuevo proyecto 2D en la última versión existente de Unity.

El primer paso que se debe realizar es adecuar la pantalla del juego para hacerla vertical y, además, preparar el entorno para que funcione correctamente en dispositivos Android. De esta forma sabremos que sobre los siguientes pasos, la aplicación estará preparada a ejecutarse en Android.

A continuación, se numeran de forma detallada las acciones realizadas, por orden de ejecución. Hay que tener en cuenta que a lo largo de esta implementación ha existido retroalimentación. Es decir, los resultados obtenidos en algunos de los pasos han llevado a nuevos cambios en pasos anteriores, con el fin de optimizar la aplicación.

1. Una vez preparada la plataforma en la que vamos a trabajar, nos encontramos ante una pantalla vertical vacía. Se decide comenzar creando la interfaz de usuario (UI) de la pantalla principal, siguiendo las directrices del diseño estudiado y asignándole elementos visuales iniciales. Se genera el primer Script 'UIController' donde se controlan los elementos de la interfaz principal.

Tras esto aparece el primer problema, dinamizar el tamaño y la ubicación de los elementos en función del tamaño y proporción de la pantalla. Tras una investigación, se consigue resolver utilizando la opción de Canvas 'Scale With Screen Size'. Además, cada elemento tendrá una ubicación dinámica en función de la ubicación del objeto padre. [7] [8]

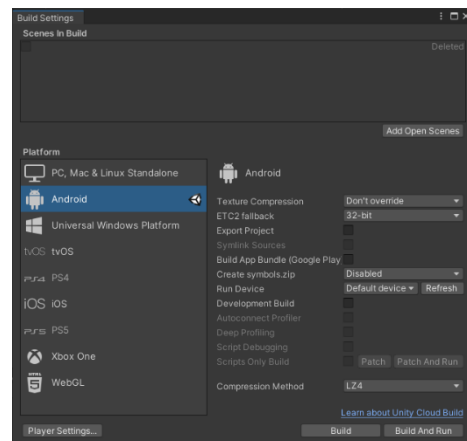


Ilustración 17. Elección de la plataforma de ejecución.

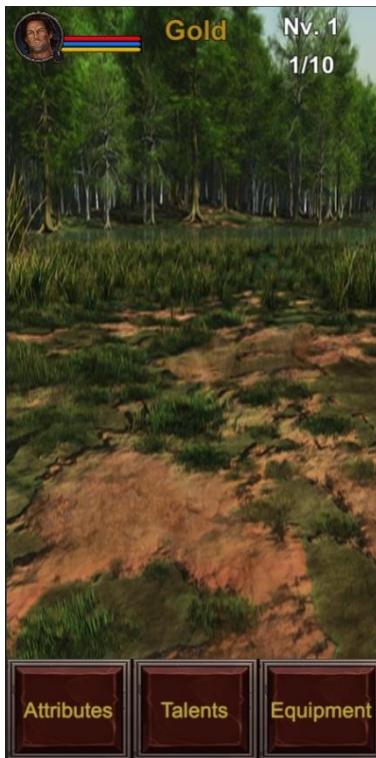


Ilustración 19. Interfaz de usuario de la pantalla principal en proporción de 18:9.

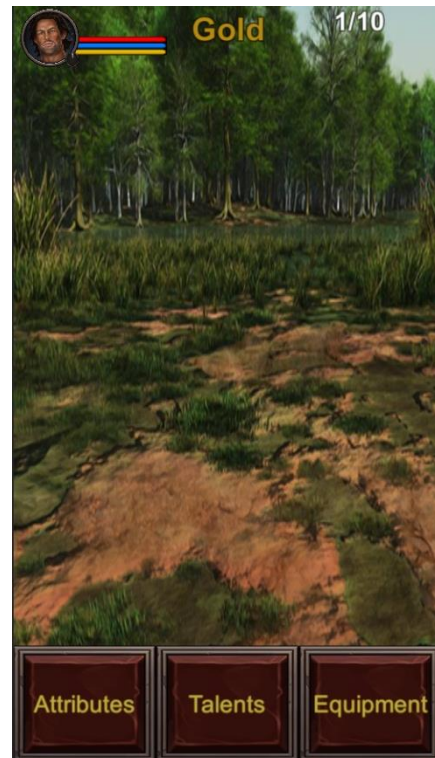


Ilustración 18. Interfaz de usuario de la pantalla principal en proporción de 16:9.

2. Una vez tenemos una visión inicial de la pantalla principal, se comienza implementando el Script principal del jugador 'Player'. Se creará e inicializará una clase Player estática (por tanto, visible desde cualquier código y escena del proyecto). Esta clase será la que contendrá todas las variables imprescindibles del jugador, que deberán ser guardadas para posteriormente ser cargadas al abrir la aplicación.
3. Como siguiente paso, se genera la zona de clickeo. Se trata de una zona que cubre la mayor parte de la pantalla principal, y que al pulsarla el personaje efectuará un ataque al enemigo actual. Entonces, se genera un objeto vacío que ocupe el espacio deseado y se le aplicará un BoxCollider2D. Generamos un nuevo script 'clickZone' donde se llamará a una función al clicar sobre el objeto. [9]
4. El siguiente paso, de gran importancia, es el guardado y carga de datos del jugador. Lo que se pretende es guardar el estado de la clase estática Player, para utilizarla en el futuro. Tras investigar, [10] [11] se opta finalmente por serializar la clase y sus subcomponentes (en pasos posteriores). Para esta utilidad, se genera un nuevo Script 'SaveSystem' que incluye una función de guardado y una de carga. Se genera un nuevo Script 'GamePlay' donde se cargarán los datos al iniciar el juego, si existen. Y si no, se creará un nuevo Player inicial.
5. Se define e implementa el objeto de enemigo, prefabricado. Tras una investigación, se decide realizar un diccionario de enemigos utilizando un archivo de extensión json, donde se ubicarán los diferentes enemigos que irán apareciendo aleatoriamente. [12] [13] Este fichero será importado al juego para disponer de su información. En este caso, se crea el Script 'JsonEnemy' y se identifican qué atributos iniciales tomarán los enemigos al crearse. Cuando se crea un enemigo, se

toma su imagen de los recursos a través de su nombre. [14] [15] El contenido del archivo .json se encuentra en el Anexo.

6. Se les añade comportamiento a las barras de salud, mana y experiencia tanto del personaje como de los enemigos. Para ello, se acuden a varias páginas webs con tutoriales para conseguirlo. [16] [17] En este paso, se genera un nuevo Script 'HealthBar' para controlar cada barra, que además cuenta con dos funciones: Set para ajustar el valor actual de la barra y SetMax para ajustar el valor máximo de la barra. Estas funciones se utilizarán en pasos posteriores.



Ilustración 20. Pantalla principal con enemigo y barras de salud.

7. Una vez se tiene la lógica del enemigo, se implementará código en los scripts ya existentes para añadir el ataque del jugador al pulsar sobre la zona de clickeo. De forma que se aplique el daño del jugador sobre el enemigo, y si este muere, se vuelva a crear un nuevo enemigo. En este caso, he investigado para realizar un buen uso de la función 'Instantiate', que crea un objeto nuevo en la ubicación deseada.
8. Se crea un Sprite prefabricado DamagePopUp, un gameObject que servirá para mostrar el daño realizado, tanto nuestro al enemigo como viceversa. En este paso se crea un nuevo Script 'DamagePopUp' para gestionar su control. [18] [19]
9. Comienza la elaboración de las interfaces de usuario 'Attributes' y 'Equipment'. Este será un paso importante, ya que se incluirán en una misma escena varias interfaces de usuario (Canvas), y se tendrán que gestionar sus usos. Al igual que la interfaz de usuario de la pantalla principal, se tendrán que preparar sus elementos para que puedan adaptarse a todo tipo de tamaño de pantallas. Junto a estas interfaces, se crearán scripts para gestionar sus elementos (UIAttributes y UIEquipment). También se definen en 'UIController' el comportamiento de los botones 'Attributes' y 'Equipment'. [20] [21] [22]



Ilustración 21. Pantalla de atributos

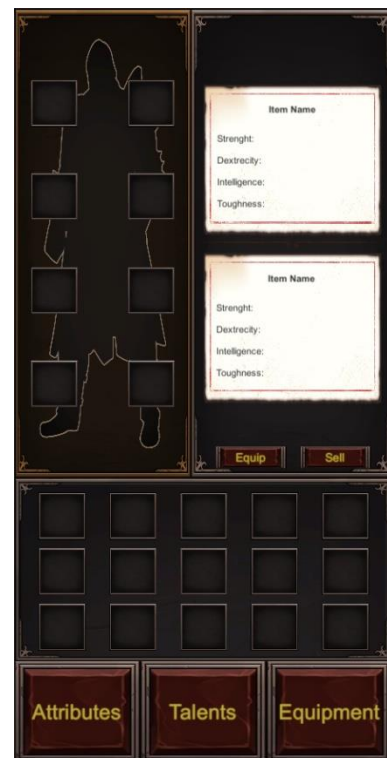


Ilustración 22. Pantalla de equipamiento.

10. Se implementa el código de 'UIAttributes'. Se trata de enseñar valores del personaje en pantalla y asignar botones para aumentar los atributos elegidos, siempre y cuando haya puntos de atributos disponibles. Al clicar sobre alguno de estos botones, se actualizarán los valores del personaje. Los atributos otorgarán los siguientes bonificadores al personaje:
 - a. Fuerza: +1 de daño de ataque, +1 de vida máxima.
 - b. Agilidad: +1 de índice de probabilidad crítica, +1% de daño crítico, +1 de índice de velocidad de ataque.
 - c. Inteligencia: +5 de maná máximo, +1 de regeneración de maná.
 - d. Dureza: +5 de vida máxima, +1 de regeneración de vida.

11. Se comienza a desarrollar el código del script UIEquipment. Antes de empezar con esto, se realiza una investigación para saber como gestionar la implementación del Equipamiento. Se encuentra y se toman como referencia algunos tutoriales. [23] [24] [25] El desarrollo del equipamiento ha sido un capítulo complejo y dificultoso en el proceso de implementación. A continuación, se detallan los aspectos destacados:
 - a. Se implementa la clase 'Item' que definirá el contenido de un objeto, así como la generación de un nuevo Item cuando sea necesario (se trata de una generación de objetos con valores aleatorios, los objetos aumentarán los atributos del personaje). El personaje contará con una Lista de Items que será su inventario y un Array de Items que será su equipamiento.

- b. Se implementa el script 'Inventory'
- c. Se rediseña la interfaz de usuario. El inventario será un 'Grid' de espacios de objeto (prefabricados). El equipamiento será un conjunto de espacios de objeto individuales, definidos con un espacio de colocación en el personaje único (esto es: casco, amuleto, armadura, guantes, arma, cinturón, botas y anillo).
- d. Se implementa el script 'Inventory Slot' que gestionará cada uno de los espacios de objeto del inventario, de manera individual.
- e. Se implementa el script 'Inventory' donde se recorrerá cada Inventory Slot para asignarle los objetos actuales del personaje (mediante un bucle).
- f. Se implementa el script 'Equipment Slot' que gestionará cada uno de los espacios de objeto del equipamiento.
- g. Una vez gestionado el control de los objetos del personaje, se implementa 'UIEquipment', donde se registran las acciones al pulsar sobre los espacios de objetos en pantalla y sobre los botones 'Equipar' y 'Vender'. Se agrega un campo booleano 'isClicked' para marcar qué objeto está seleccionado en cada momento.

- i. Seleccionar un objeto del inventario: El espacio del objeto seleccionado cambia de color, se seleccionará automáticamente el objeto equipado equivalente. Se mostrará información de cada objeto en los cuadros de descripción.
- ii. Seleccionar un objeto del equipo: El espacio del objeto seleccionado cambia de color, se mostrará información del objeto equipado en el cuadro de descripción. Se deselecciona cualquier objeto del inventario seleccionado.
- iii. Botón equipar: Si hay algún objeto del inventario seleccionado, éste pasará a ser equipado. Si ya hay un objeto equipado, estos se intercambian; si no lo hay, pasa del inventario al equipo.
- iv. Botón vender: Si hay algún objeto del inventario seleccionado, se destruirá para ganar oro en función del poder del objeto.



Ilustración 21. Pantalla de equipamiento completa.

- 12. Se implementa el mecanismo del auto ataque, tanto para el personaje como para el enemigo. Se implementan las funciones de definición o ecuaciones de atributos secundarios a partir de los atributos primarios. [26] [27] [28] Se trata de mantener los atributos secundarios afinados a medida que los atributos primarios van aumentando. El objetivo es conseguir un crecimiento exponencialmente negativo. Esto es, que crezca en mayor cantidad al principio y conforme suben los atributos

primarios, el crecimiento de atributos secundarios se vaya viendo mermado, traduciéndose en una mayor dificultad. Para ello, se investigan actuaciones comunes en otros juegos y se realizan varias pruebas sobre cada uno de ellos:

- a. Velocidad de ataque: en función de la destreza (eje y). Segundos por Ataque. Al principio se gana mucha velocidad de ataque, pero luego la ganancia va disminuyendo.

$$Attack\ Speed = 3 + 3 * \frac{1 - Dex}{Dex + 4}$$

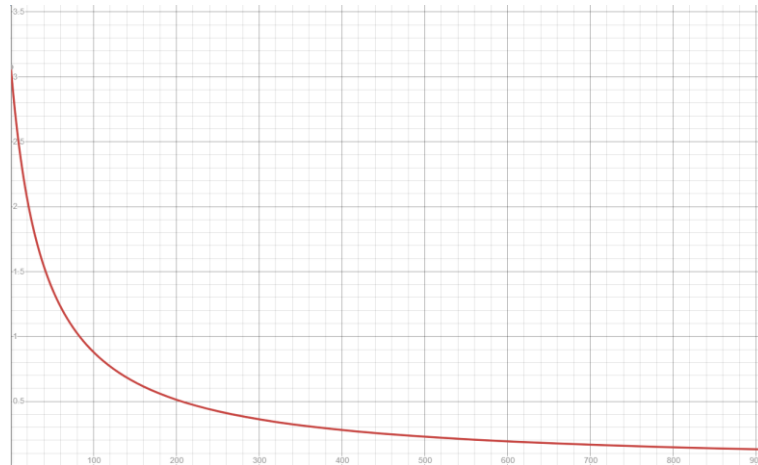


Ilustración 22. Gráfica de velocidad de ataque.

- b. Experiencia máxima para subir de nivel (eje y), por nivel (eje x). Cada vez cuesta más experiencia subir de nivel, exponencialmente.

$$Max.\ Exp. = 0.5 * lvl^2 + lvl * 10$$

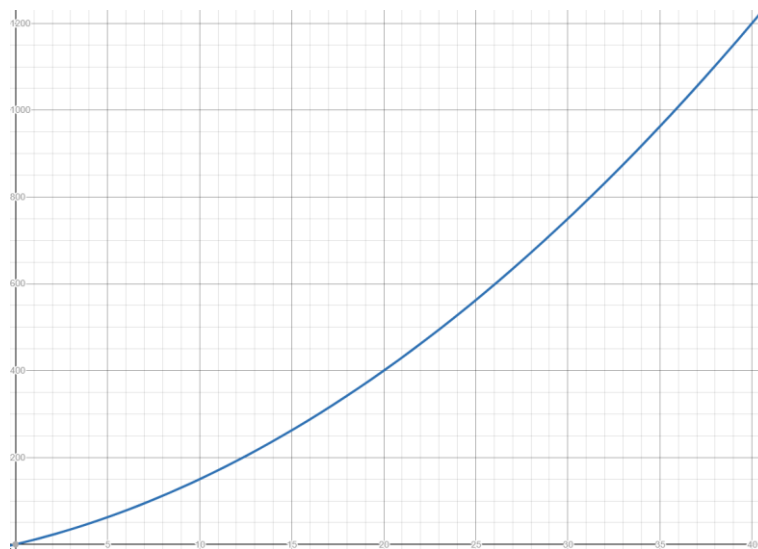


Ilustración 23. Gráfica de la experiencia máxima por nivel.

- c. Absorción de daño mediante defensa (eje x, en porcentaje de daño no absorbido 0-1), en función de la dureza (eje y)

$$\text{Daño} = \text{daño total} * \frac{500}{500 + \text{defensa}}$$

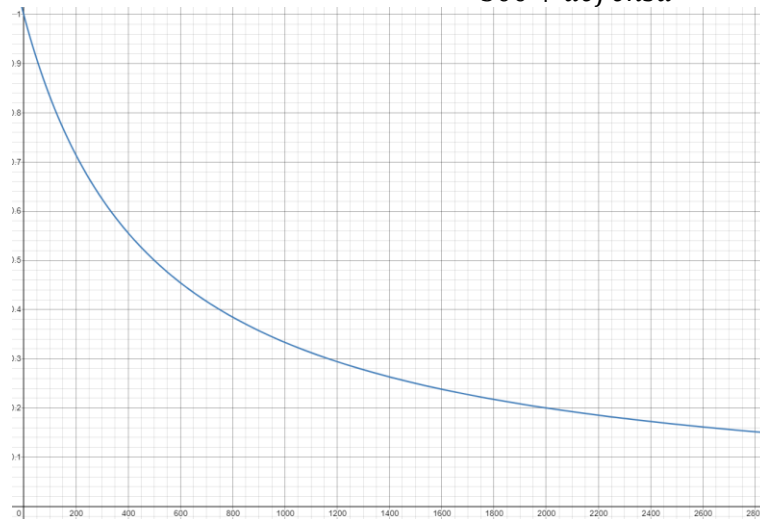


Ilustración 24. Gráfica de daño absorbido por defensa.

d. Índice de ataque crítico (eje y, en probabilidad de 0-1), en función de la destreza.

$$\text{Crit. Prob.} = \frac{\text{Dex}}{\text{Dex} + 300}$$

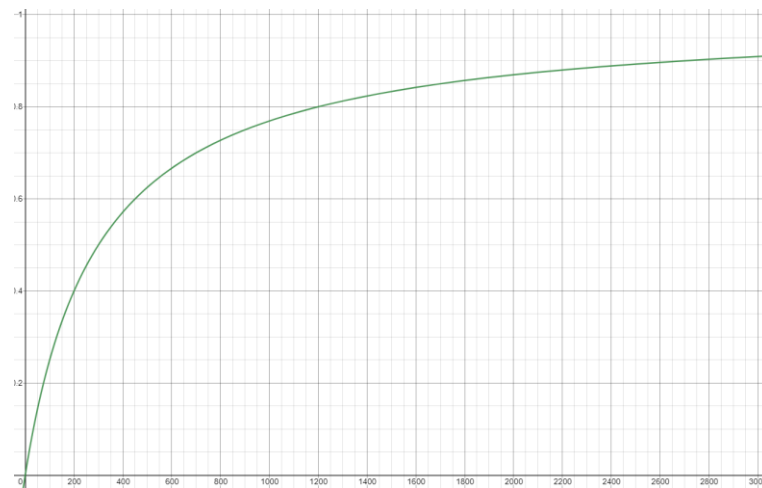


Ilustración 25. Gráfica de Probabilidad de Crítico

13. Implementación de la interfaz de usuario y código de la recompensa al volver al juego. Como parte fundamental en este tipo de juegos, calcularemos el tiempo transcurrido entre el último uso de la aplicación y la entrada a la misma, de forma que se logren recompensas. En este caso, las recompensas son:

- a. 10 * Nivel del mapa por cada minuto transcurrido.
- b. 1 objeto por cada 10 minutos transcurrido.

14. Se elabora la interfaz de usuario de Talentos, junto con el código que gestiona sus controles 'UITalents'. Además, se modifica el resto de códigos para que los talentos tengan efectos. En cada nivel se podrá subir un punto a un talento, hasta un máximo

de 20 por cada uno. Existirán 9 talentos, serán 3 asociados a fuerza, 3 a destreza y 3 a inteligencia:

- a. Talentos de Fuerza:
 - i. Furia: + 1% de daño por cada punto.
 - ii. Robo de vida: +1% de robo de vida (1% del daño realizado aplica como sanación).
 - iii. Rabia: +2% de daño y +1% de daño recibido por cada punto.
- b. Talentos de Destreza:
 - i. Puntería: +1% de probabilidad crítica y de daño crítico por cada punto.
 - ii. Acrobacias: +1% de evadir ataques por cada punto.
 - iii. Rapidez: +1% de velocidad de ataque por cada punto.
- c. Talentos de Inteligencia:
 - i. Bola de fuego: Gasta 1 de maná por punto en cada ataque para aumentar su daño un 10% por cada punto.
 - ii. Mana Shield: +2% del daño recibido va al maná en vez de a la vida por cada punto.
 - iii. Claridad: +1% de regeneración de maná por cada punto.

Como se puede ver, se ha elegido que los talentos aumenten estadísticas en porcentaje, de forma que aporten más valor en partidas avanzadas, donde los atributos son mayores.

15. Se agregan los sonidos y la música al juego. Para ello se crea un nuevo script 'AudioManager' que creará un vector de Audiosources (uno por cada sonido) y funciones Play y Stop. Modificamos el resto de códigos para reproducir los sonidos en los momentos deseados. Estos sonidos son:
- a. Música de fondo.
 - b. Efecto al atacar (3 distintos).
 - c. Efecto al acabar con un enemigo.
 - d. Efecto al subir de nivel.
 - e. Efecto al esquivar un ataque.

16. Por último, se crea una escena inicial de título y presentación del juego. Pulsando sobre cualquier parte de la pantalla se accederá a la escena principal, creando un nuevo script 'Initial'.

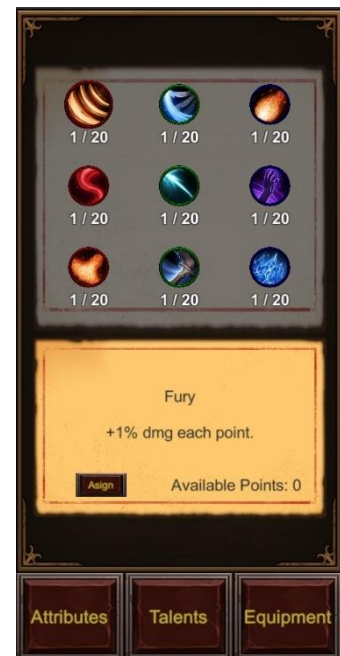


Ilustración 26. Pantalla de talentos.

3.2. Recursos

Para la realización de este proyecto, he decidido comprar dos packs de recursos desde la página web [HumbleBundle](https://www.humblebundle.com), consiguiendo una gran variedad de imágenes y sonidos con descuento. Humble Bundle es una web que ofrece paquetes de juegos, herramientas software y libros digitales a un precio reducido. Además, el precio a pagar en cada paquete es libre, y puedes distribuir lo que pagas a diferentes asociaciones benéficas, los desarrolladores de los productos y la propia página web. El resto de

recursos utilizados vienen de packs gratuitos o imágenes sueltas provenientes de librerías de recursos gratuitos. Finalmente, los recursos utilizados han sido:

1. Imágenes:

- a. UI: Assets gratuitos de Unity Store 'Fantasy Free UI' y 'GUI Parts'



Ilustración 27. Ejemplo de elementos UI utilizados en la aplicación.

- b. Enemigos: Se han utilizado varios de los recursos comprados en Humble Bundle.



Ilustración 28. Ejemplo de imágenes de enemigos utilizados en la aplicación.

- c. Objetos: Se han utilizado imágenes de Assets gratuitas en la Asset Store de Unity 'Basic RPG Icons' y 'RPG inventory icons'.



Ilustración 29. Imágenes de ejemplo de objetos utilizado en la aplicación.

2. Sonidos

- a. Música: Se han utilizado melodías de fondo obtenidas de los recursos comprados en Humble Bundle.
- b. Efectos: Se han utilizado diferentes sonidos descargados de librerías online gratuitas.

3.3. Pruebas

Tras la preparación del entorno en el que se desarrolló el propio juego para ajustar su funcionamiento en dispositivos móviles, Unity permite realizar ejecuciones virtuales. Además, el motor cuenta con una ventana de consola donde podremos mostrar desde código los valores que nos interese comprobar.

La ejecución virtual de Unity es una potente herramienta en la que también podremos cambiar el tamaño de la pantalla del dispositivo, o incluso cambiar el valor de los componentes de los objetos que se encuentren en pantalla en tiempo real. Este recurso se ha venido utilizando con asiduidad a lo largo del proceso de implementación del proyecto.

Testeos en dispositivos móviles: Se han realizado varias pruebas en mi dispositivo móvil personal. Para ello, copié el archivo .apk al dispositivo mediante cable USB e instalé la aplicación. Las pruebas confirmaron el buen funcionamiento que el juego tenía dentro de las pruebas de Unity.

Por otro lado, el proyecto ha sido llevado a la plataforma de Google Play, desgraciadamente, la aplicación necesita ser revisada antes de proceder a su uso. Por motivos de retrasos en el tiempo de revisión, no ha sido posible incluir el proyecto a través de esta plataforma (El proyecto lleva más de dos semanas en espera).

4. Cumplimentación de objetivos

A lo largo del proceso de implementación de este proyecto, han aparecido algunas dificultades que finalmente no han permitido cumplir con las expectativas de funcionalidades del proyecto. La carga de trabajo estimada para el desarrollo de la aplicación ha sido subvalorada, además de encontrarse con algunas dificultades en el camino.

La mayor dificultad encontrada fue el funcionamiento del equipamiento, que conlleva un mayor tiempo de trabajo del esperado. Seguidamente, la búsqueda de ecuaciones ha sido una tarea ardua para dar con una solución razonable a lo largo del tiempo de uso de la aplicación.

Debido a estos cambios de horas trabajadas en algunas de las funciones, se decide realizar modificaciones en las siguientes funciones a implementar, con el objetivo de obtener un producto que sirva como cimiento sobre el cuál ir realizando mejoras a futuro.

5. Conclusiones

En este trabajo fin de máster se ha elaborado el esqueleto de un videojuego idle con elementos de progresión característicos de juegos RPG. Este esqueleto servirá como base para un juego con gran potencial. A partir de aquí se podrán agregar muchísimas mejoras, hacerlo competitivo y finalmente, monetizarlo.

Con el proyecto he aprendido a los pasos que se deben llevar a cabo para realizar una aplicación de gran envergadura. Se necesitan horas de trabajo y dedicación si queremos hacer realidad lo que antes era solo una idea, especialmente si es una única persona la que lo implementa.

Aunque no se han podido implementar todas y cada una de las funciones mencionadas en la etapa de planificación y diseño, se presenta un juego que puede ser un reto por sí solo. ¿Hasta que nivel seremos capaces de llegar?

A líneas de futuro, este juego tiene muchas posibilidades para mejorar. Hay un mundo de características adicionales a agregar. Y muy probablemente iré realizando de forma personal. Por listar varias mejoras que se me puedan ocurrir (algunas ya mencionadas anteriormente):

- Clasificación online del mayor nivel de mapa alcanzado por los usuarios en el juego: Agregar competitividad en la aplicación hará que los usuarios la utilicen muchísimo más con el objetivo de superar a sus rivales (otros usuarios) que pueden tratarse de amigos en la vida real.
- Darle una utilidad al oro ganado: Por ejemplo, mejorando el equipo, comprar cofres con nuevo equipamiento, resetear puntos de talento y/o de atributos, aleatorizar los valores de una pieza de equipo, etc..
- Habilidades activas: Algo que se echa en falta en todo juego RPG, agregar a los talentos habilidades activas, que sean elegibles desde la pantalla principal y que al clicar se active un efecto adicional, beneficioso para el personaje.
- Ramas de talentos: Ahora mismo cualquier talento es elegible, el objetivo real sería ir desbloqueando nuevos talentos a medida que se eligen los que están desbloqueando.
- Mejora en los equipamientos: hacer más complejos los objetos, que puedan aportar beneficios diferentes que no sean solo atributos. También añadir la posibilidad de que aparezcan objetos únicos y poderosos, que tienen sus modificadores establecidos.

Y muchos más...

6. Glosario

En este apartado se encuentran las siglas o definiciones que pueden no haberse definido con claridad y ser desconocidas por el lector, o simplemente para refrescar sus significados:

1. Idle: O inactividad, se trata de un tipo de videojuegos llamado así por que se consigue una sensación de progresión aún sin utilizar la propia aplicación, ganando recompensas al entrar en la aplicación en función del tiempo sin haberla utilizado.

Además, muchas veces, dentro del propio juego se consiguen recursos sin necesidad de realizar acciones.

2. RPG: O Role Playing Game, es un género de videojuegos donde el jugador controla las acciones de un personaje (o de diversos miembros de un grupo) inmerso en algún detallado mundo.
3. TFM: Trabajo Final de Máster.
4. Front-end: En diseño de software, el front end es la parte del software que interactúa con los usuarios, se trata de la capa de presentación.
5. Back-end: En diseño de software, el back end es la parte que procesa la entrada desde el front end, se trata de la capa de acceso de datos.
6. Humble Bundle: son paquetes de juegos, a menudo desarrollados por estudios independientes, que son distribuidos de forma en línea y cuyo precio es marcado por el propio comprador, pudiendo elegir la cantidad a pagar por los juegos ofrecidos, sea cual sea la cifra.
7. Assets: En juegos, representa cada uno de los elementos que componen el juego. Al estar en una tienda, se refiere a un paquete donde se incluyen uno o más de estos elementos.
8. Unity: es un motor de videojuego multiplataforma creado por Unity Technologies. Unity está disponible como plataforma de desarrollo para Microsoft Windows, Mac OS, Linux. La plataforma de desarrollo tiene soporte de compilación con diferentes tipos de plataformas
9. Unity Store: Plataforma de Unity desde la que se pueden comprar y descargar numerosos Assets para el sistema Unity.

7. Bibliografía

- [1] «Google Play,» [En línea]. Available: <https://play.google.com/store>. [Último acceso: 1 3 2021].
- [2] M. Martínez, «'Idle games': videojuegos en los que se mira la pantalla, y ya,» 14 11 2018. [En línea]. Available: <https://www.nobbot.com/pantallas/idle-games-videojuegos-mira-pantalla/>. [Último acceso: 3 5 2021].
- [3] M. Blanco, «Los 4 idle games más adictivos que debes jugar,» 1 4 2019. [En línea]. Available: <https://www.geekno.com/los-idle-games-mas-adictivos-que-debes-jugar.html#:~:text=%20Los%20%20juegos%20idle%20m%C3%A1s%20populares%20en,of%20Legends,%20te%20permite...%20%20Spaceplan%20More>. [Último acceso: 5 3 2021].
- [4] «Unity,» [En línea]. Available: <https://unity.com/es>. [Último acceso: 3 3 2021].
- [5] T. Rodríguez, «Métodos aplicables para el desarrollo de aplicaciones móviles,» 29 Septiembre 2011. [En línea]. Available: <https://www.genbeta.com/desarrollo/metodos-aplicables-para-el-desarrollo-de-aplicaciones-moviles>. [Último acceso: 4 3 2021].
- [6] «Desarrollo ágil de software,» [En línea]. Available: https://www.ecured.cu/Desarrollo_%C3%A1gil_de_software. [Último acceso: 4 3 2021]
- [7] «Scale Canvas and UI with Screen Size,» [En línea]. Available: <https://unitycoder.com/blog/2020/03/07/scale-canvas-and-ui-with-screen-size/>.
- [8] «How to fit screen size for every resolution Unity,» 2017. [En línea]. Available: <https://youtu.be/X4AdT5cKmZQ>.
- [9] «How to make a sprite clickable?,» 2015. [En línea]. Available: <https://stackoverflow.com/questions/28146509/how-to-make-a-sprite-clickable>.
- [10] «Como Guardar y Cargar el progreso del jugador en Unity,» 2014. [En línea]. Available: <https://gamedevelopment.tutsplus.com/es/tutorials/how-to-save-and-load-your-players-progress-in-unity--cms-20934>.
- [11] «SAVE & LOAD SYSTEM in Unity,» 2018. [En línea]. Available: https://youtu.be/XOjd_qU2ldo.
- [12] «Instantiating Prefabs at run time,» [En línea]. Available: <https://docs.unity3d.com/Manual/InstantiatingPrefabs.html#:~:text=To%20instantiat e%20a%20Prefab%20at%20run%20time%2C%20your,Prefab%20you%20want%20to%20use%20in%20the%20Inspector..>
- [13] «Simple JSON to Unity For Object Data,» 2017. [En línea]. Available: <https://www.jonathanyu.xyz/2017/02/27/simple-json-to-unity-for-object-data/>.
- [14] «How to load image from resource folder and set it to source image of UI Image in unity5,» 2015. [En línea]. Available: <https://stackoverflow.com/questions/31719865/how-to-load-image-from-resource->

[folder-and-set-it-to-source-image-of-ui-image-in.](#)

- [15] «(SOLVED) Unable to load Sprites from Resources folder,» 2017. [En línea]. Available: <https://forum.unity.com/threads/solved-unable-to-load-sprites-from-resources-folder.524807/>.
- [16] «How to make a HEALTH BAR in Unity!,» 2020. [En línea]. Available: https://www.youtube.com/watch?v=BLfNP4Sc_iA.
- [17] «Game Dev: How to Make Health Bars in Unity From Beginner to Advanced,» 2018. [En línea]. Available: <https://medium.com/swlh/game-dev-how-to-make-health-bars-in-unity-from-beginner-to-advanced-9a1d728d0cbf>.
- [18] C. Monkey, «How to make Damage Popup Text (Unity Tutorial),» 2019. [En línea]. Available: https://www.youtube.com/watch?v=iD1_JczQcFY.
- [19] Dennis, «Unity Damage Popup (Numbers | Text) Tutorial | Over Time Fade Out | Beginner Friendly | Clean,» 2019. [En línea]. Available: <https://youtu.be/iQo02TvVXaU>.
- [20] «how to enable a gameobject?,» 2008. [En línea]. Available: <https://forum.unity.com/threads/how-to-enable-a-gameobject.10372/>.
- [21] «How to enable/ disable GameObject,» 2014. [En línea]. Available: <https://gamedev.stackexchange.com/questions/81322/how-to-enable-disable-gameobject>.
- [22] «Hide/Unhide Canvas in Unity AR Project,» 2020. [En línea]. Available: <https://stackoverflow.com/questions/63155886/hide-unhide-canvas-in-unity-ar-project>.
- [23] Brackeys, «Lista de Reproducción: How to make an RPG in Unity. Vídeos 4, 5, 6 y 7,» 2018. [En línea]. Available: <https://www.youtube.com/playlist?list=PLPV2Kylb3jR4KLGCCAcWQ5qHudKtYeP7>
- [24] GameGrind, «Inventory System in Unity Tutorial (Part 4) - Unity3D C# Tutorial,» 2014. [En línea]. Available: <https://www.youtube.com/watch?v=98nKQ15of6E>.
- [25] «Array.Find<T>(T[], Predicate<T>) Method,» [En línea]. Available: <https://docs.microsoft.com/en-us/dotnet/api/system.array.find?view=net-5.0>.
- [26] «Puntos de Experiencia,» [En línea]. Available: https://habitica.fandom.com/es/wiki/Puntos_de_Experiencia.
- [27] «HOW DO ARMOR AND MAGIC RESISTANCE WORK IN LOL?,» 2017. [En línea]. Available: <https://boosteria.org/guides/armor-magic-resistance-work-league-legends>.
- [28] «Formulas in Combat Design,» 2020. [En línea]. Available: <https://www.liangzhengluo.com/blog/formulas>.

8. Anexos

En este capítulo se incluirán apartados adicionales de carácter autocontenido.

8.1. Contenido del archivo de enemigos enemies.json

```
{
  "enemy": [
    {
      "id": 1,
      "name": "Skeleton",
      "hp": 10,
      "dmg": 1
    },
    {
      "id": 2,
      "name": "Ghost",
      "hp": 8,
      "dmg": 1.2
    }
  ]
}
```

8.2. Manual de instalación

8.2.1. Abrir el proyecto en Unity

1. Descargar e instalar la última versión del motor Unity. ([Link](#))
2. Una vez tenemos el programa preparado, ejecutar Unity Hub.
3. Instalar la versión del programa necesario para la ejecución de nuestro proyecto. Nuestro proyecto se ha desarrollado en la última versión disponible de Unity actual. 2021.1.6f1. También será necesario agregar los módulos de 'Android Build Support'.
4. Una vez instalada la versión de Unity asociada al proyecto. Acudir a la sección de proyectos 'Projects' y clicar en ADD. Aparecerá una ventana para buscar archivos, elegir la carpeta del proyecto 'TFM'.
5. Tras esto, ya tendremos disponible nuestro proyecto para ser ejecutado a través de Unity.

En mi caso, he elegido del programa Visual Studio Code para implementar todo el código. En las opciones de Unity se puede especificar que se utilice este programa por defecto a la hora de utilizar nuestros scripts.

8.2.2. Ejecutar el proyecto en un dispositivo móvil.

1. Conectar el dispositivo móvil al PC mediante USB para copiar el archivo 'TFM_Build.apk'.
2. Este archivo es un instalador directo de nuestro proyecto. Por tanto, una vez esté situado en el dispositivo móvil se podrá ejecutar su instalación. Hay que tener en

- cuenta que se trata de un desarrollador desconocido, por tanto, te preguntará si realmente deseas instalar esta aplicación. Pulsar 'Instalar de Todas Formas'
3. Ya tendremos nuestro proyecto 'TFM' instalado en el dispositivo, como una aplicación habitual.

8.2.3. Ejecutar el proyecto a través de Google Play.

1. Recientemente, he tratado de agregar la aplicación a Google Play. A través de la plataforma 'Google Play Console', que gestiona el proceso para subir aplicaciones a Google Play.
2. Actualmente, se ha subido el proyecto en un estado de Prueba interna, donde una serie de testadores elegidos a través de una lista de direcciones de correo podrán instalar y probar la aplicación a través de Google Play.
3. Sin embargo, antes de que esta aplicación sea accesible a los testadores, deberá ser revisada por el equipo de Google Play. Desgraciadamente, la misma página te advierte que el tiempo de revisión es más largo de lo habitual.
4. Hace ya una semana de esto, y la aplicación se encuentra sin revisar.

8.3. Manual de usuario

En este proyecto se ha desarrollado un juego especialmente para dispositivos móviles Android (aunque gracias a Unity, se podría adaptar a otros entornos). El objetivo del juego es conseguir llegar al mayor nivel de mapa posible, enfrentándose cada vez a enemigos más poderosos. Para poder avanzar de niveles, el personaje dispone de atributos (fuerza, destreza, inteligencia y dureza), que podrán aumentarse al subir su nivel y equiparse objetos que iremos ganando conforme derrotemos enemigos.

Al llegar a la pantalla principal por primera vez, aparecemos en el nivel uno y ya tendremos nuestro primer monstruo atacándonos. Pulsa sobre cualquier espacio de la pantalla, salvo los botones inferiores, para realizar un ataque al enemigo. Por cada pulso efectuarás un ataque, así que ¡aprovecha! Los enemigos te atacarán cada segundo.

Según derrotas enemigos, irás ganando experiencia para subir de nivel y ganando objetos, que serán más poderosos a medida que avances en los niveles del mapa. Por cada nivel de personaje que consigas, tendrás 5 puntos para distribuir en tus atributos.

Para aumentar tus atributos, pulsa el botón 'Atributos' para acudir a la pantalla de atributos, en ella podrás ver las estadísticas de tu personaje, y como irán aumentando conforme a tus atributos. Para utilizar tus puntos disponibles dispones de un botón '+' por cada atributo.

Por otro lado, puedes acceder a la pantalla de equipamiento pulsando el botón 'Equipment'. En ella encontrarás los objetos obtenidos al derrotar enemigos en la parte inferior 'Inventario'. Si pulsas sobre cada uno de ellos podrás equiparlos o venderlos con los botones asociados. También al pulsar sobre los objetos se mostrarán qué atributos te sumarán al estar equipados.

Se recomienda enormemente ir vendiendo objetos del inventario que no quieras usar, de forma que siempre tengas espacio en el inventario para nuevos objetos.

Ten cuidado a lo largo de la partida, ya que los enemigos irán ganando poder a medida que avanzas de fase. Si los enemigos logran acabar contigo, perderás experiencia y regresarás a un nivel de mapa inferior.

Además, si no estás jugando, ¡Sigues ganando recompensas! Al entrar de nuevo en el juego, recibirás recompensas en función del tiempo que hayas estado desconectado y del nivel de mapa en el que te hayas quedado.

