

WORLD OF WARS

DESARROLLO DE APLICACIONES ANDROID

Autor: Albert Fabregat Hernando

Consultor: Marc Domingo Prieto

WORLD OF WARS

DESARROLLO DE APLICACIONES ANDROID

INDICE

Descripción del proyecto	4
Objetivos	4
Objetivos mínimos.....	4
Objetivos deseables.....	4
Tecnología a utilizar.....	5
Planificación	6
Riesgos del proyecto	8
Diagrama de Gantt.....	9
Herramientas y dispositivos utilizados	10
Usuarios y contexto de uso	10
Métodos de indagación	11
Resultados.....	13
Perfil de usuarios identificados	15
Contexto de uso	16
Análisis de tareas.....	16
Disseño conceptual.....	18
Personas	18
Escenarios de uso.....	20
Flujos de interacción	22
Prototipaje	23
Sketches.....	23
Prototipo de alta fidelidad	25
Inicio.....	25
Nueva partida	26



Continuar partida	26
Opciones.....	27
Mapa del mundo.....	27
Inventario	28
Tienda.....	28
Dentro juego	29
Evaluación.....	30
Test de usuarios.....	30
Implementación.....	32
Arquitectura	32
Referencia de archivos.....	35
Utils	36
Managers.....	36
Entities	37
Screen.....	39
Conclusiones.....	43
Fuentes de información	44
Websites.....	44
Libros	44

INDICE DE ILUSTRACIONES

Ilustración 1: Gráfica de OS Móvil.....	13
Ilustración 2: Gráfica de tipo de jugador	13
Ilustración 3: Gráfica de cooperativo vs competitivo	14
Ilustración 4: Gráfica de características mejor valoradas.....	14
Ilustración 5: Gráfica de características más atractivas	14
Ilustración 6: Gráfica de géneros	14
Ilustración 7: María González.....	18
Ilustración 8: Juan Fernández	19
Ilustración 9: Miguel Jacobo	20
Ilustración 10: Flujo de interacción.....	22
Ilustración 11: Sketch Game-Play	23
Ilustración 12: Sketch menús 1	23
Ilustración 13: Sketch menús 2	24
Ilustración 14: Prototipo pantalla inicial.....	25
Ilustración 15: Prototipo nueva partida.....	26
Ilustración 16: Prototipo continuar partida	26
Ilustración 17: Prototipo opciones	27
Ilustración 18: Prototipo mapa.....	27
Ilustración 19: Prototipo inventario	28
Ilustración 20: Prototipo tienda	28
Ilustración 21: Prototipo Game-Play.....	29
Ilustración 22 Diagrama de la lógica de ataque.....	34
Ilustración 23 Diagrama de 'utils'.....	36
Ilustración 24 Diagrama de los Managers.....	37
Ilustración 25 Diagrama de las Entidades	37
Ilustración 26 Diagrama de las pantallas	39
Ilustración 27 Captura del menu	40
Ilustración 28 Captura de las Opciones	40
Ilustración 29 Captura del mapa	41
Ilustración 30 Captura de la tienda.....	41
Ilustración 31 Captura del Juego en una misión.....	42

DESCRIPCIÓN DEL PROYECTO

La idea principal de *World Of Wars* es la creación de un videojuego de acción para plataformas *Android* basándose en los estilos *shoot em'up arcades* antiguos pero adaptándolos a los nuevos tiempos y nuevas formas de juego que existen en la actualidad.

A grandes rasgos, se podrá considerar *World Of Wars* como un juego de aviones donde la destrucción y el rescate serán las principales herramientas para superar los niveles.

OBJETIVOS

Objetivos mínimos

El principal objetivo del proyecto es desarrollar una versión en la que se pueda disfrutar de la experiencia del juego. Para ello será necesario que exista una pantalla de inicio donde se pueda iniciar una nueva partida o cargar una partida anterior. Este objetivo mínimo se considerará logrado si se consigue que un jugador pueda interactuar con el personaje principal (avión), pueda atacar a los intentando recibir el mínimo daño posible de los enemigos hasta poder llegar al punto seguro.

Otro objetivo indispensable, es poder mejorar el avión principal mediante la compra de objetos y mejoras para poder hacer frente a los enemigos que se vuelven superiores y más duros durante el progreso del juego.

Objetivos deseables

El principal objetivo deseado sería realizar un sistema de logros, donde se puedan mostrar las mejores puntuaciones. Otro gran objetivo a realizar, es poder generar una línea histórica no lineal mediante misiones del juego para dotarle de una historia y no hacerlo aburrido.

TECNOLOGÍA A UTILIZAR

Para la realización del proyecto se han barajado diferentes tecnologías a utilizar. Al ser un proyecto orientado a dispositivos móviles se ha realizado una búsqueda de diferentes motores o librerías para la creación de este, pero con los siguientes requisitos:

- ✚ Multiplataforma
- ✚ Lenguaje C/C++ o Java.
- ✚ *OpenSource*
- ✚ Motor 2D de alto rendimiento

Se han encontrado básicamente dos alternativas que cumplen los requisitos deseados:

✚ **LibGDX**

Es un *framework* de desarrollo de juegos programado en Java que proporciona una API unificada multiplataforma. El *framework* proporciona un entorno para la creación rápida de prototipos y de altas iteraciones. Se puede programar casi en su totalidad sin que sea necesario desplegar la aplicación en un terminal real o uno virtual, ya que se puede desarrollar de forma nativa en el PC (Windows/Linux/OSx) y probar todo el código antes de desplegarlo a la plataforma final, en este caso Android.

✚ **Cocos2dx**

Es un *framework* multiplataforma para la creación de juegos 2D, libros interactivos, demostraciones o cualquier aplicación gráfica. Está basado en cocos2d-iphone, pero en vez de utilizar Objective-C, está programado en C++. Funciona en iOS, Android, OSx, Windows, Linux, Emscripten, Google Native Client, BlackBerry Tizen. Es rápido, libre, fácil de usar y dispone de una gran comunidad detrás.

Con ambos *frameworks* he podido realizar pequeños juegos y demostraciones. Pero finalmente para este proyecto he decidido utilizar Cocos2dx porque dispongo de experiencia programando en C++ y me siento mucho más cómodo utilizando este lenguaje en vez de Java.

Como para la realización del proyecto utilizaré diferentes máquinas, el proyecto deberá compilar y funcionar correctamente en los tres sistemas operativos (Windows, Linux, OSX) con el menor esfuerzo posible e intentando no realizar código dependiente de plataforma. Para ello, se intentará utilizar las menores librerías externas posibles, así como el estándar C++11 de la iteración del lenguaje de programación C++.

PLANIFICACIÓN

1. Análisis previo

- ✓ **Objetivo:** Definir el tipo de juego que se desea desarrollar y seleccionar las tecnologías a usar.
- ✓ **Finalidad y riesgos:** La finalidad de esta fase es servir como punto de partida. Es la primera fase obligatoria del proyecto.

2. Diseño del gameplay

- ✓ **Objetivo:** Se realizará un diseño inicial de la base del juego.
- ✓ **Finalidad y riesgos:** La finalidad es conseguir una mecánica de juego que sea divertida y simple para usar desde un dispositivo móvil. Es posible que durante la fase del prototipo se detecten deficiencias del diseño y se tenga que volver a retomar de nuevo esta fase.

3. Análisis técnico:

- ✓ **Objetivo:** Análisis donde definiremos las diferentes partes a implementar.
- ✓ **Finalidad y riesgos:** El objetivo es definir los aspectos más técnicos del desarrollo.

4. Diseño técnico:

- ✓ **Objetivo:** Elaboración de diagramas técnicos necesarios para la fase de desarrollo.
- ✓ **Finalidad y riesgos:** Los diagramas son fundamentales para la fase de implementación. Una deficiencia en el diseño técnico puede acarrear numerosos problemas. Hay que prestarle una gran atención.

5. Prototipado

- ✓ **Objetivo:** Será necesario realizar un primer prototipo para poder comprobar que todo lo diseñado es viable y no se haya dejado ningún detalle.
- ✓ **Finalidad y riesgo:** Asegurar un diseño sólido, divertido y jugable.

6. Pruebas

- ✓ **Objetivo:** Comprobar que el prototipo es robusto y con los menores errores posibles.
- ✓ **Finalidad y riesgos:** Hay que asegurarse que el juego no tiene fallos o errores que hagan que la experiencia del juego sea nefasta o imposible de jugar.

7. Desarrollo

- ✓ **Objetivo:** En esta fase se implementará el juego por completo, validando el prototipo y extendiendo la base del juego para dotarla de vida. Se definirán diferentes sub-tareas para dicha fase.
- ✓ **Finalidad y riesgos:** Es bastante probable que durante esta fase, ocurran muchos errores de desarrollo y que se deba de prestar especial atención para intentar corregirlos lo antes posible y nos permita avanzar sin problemas.

8. Testing

- ✓ **Objetivo:** Comprobar que el juego funciona correctamente y no existen errores. Para ello se realizará una búsqueda de errores y se intentará realizar *tests funcionales*.
- ✓ **Finalidad y riesgos:** Localizar todos los errores posibles para disponer de un proyecto sólido y estable.

9. Documentación

- ✓ **Objetivo:** La elaboración de la documentación final.
- ✓ **Finalidad y riesgos:** La finalidad será realizar un manual de usuario.

RIESGOS DEL PROYECTO

Se han planteado los siguientes riesgos del proyecto:

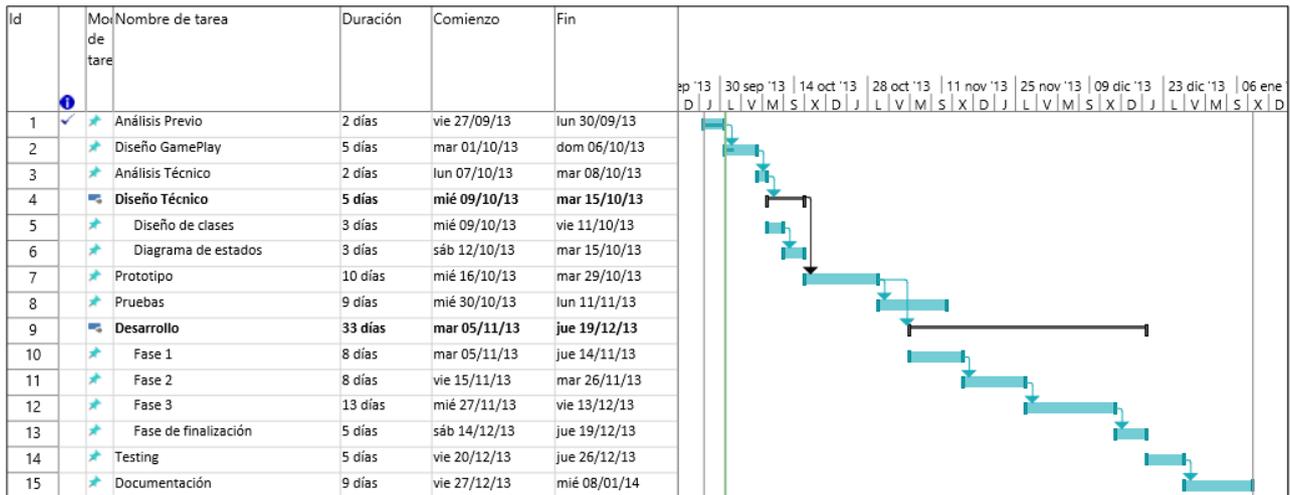
- ✚ Utilización de una versión no muy estable del framework seleccionado.
- ✚ Calendario muy ajustado para realizar todas las tareas deseadas.
- ✚ Problema a la hora de diseñar gráficamente los elementos visuales ya que no se dota de mucha experiencia y soltura en ello.
- ✚ Poca disponibilidad horaria.

Para poder solucionar estos riesgos en el caso de que ocurrieran, se debería de intentar realizar lo antes posible un primer prototipo para comprobar que el *framework* cumple los requisitos deseados y no ocurren problemas. En el caso del calendario muy ajustado se deberá seguir al pie de la letra la planificación y en el caso que surja algún problema intentar priorizar las tareas.

El problema con los elementos visuales, es posible encontrar por internet multitud de recursos gratuitos con licencias *Creative Commons*, e incluso si fuera necesario adquirirlos pagando en webs especializadas de venta de recursos para programadores de videojuegos.

Por último la poca disponibilidad horaria se podrá minimizar intentando trabajar en todos aquellos equipos informáticos en los que se tenga acceso, en los tiempos muertos, ya sea en el trabajo, mientras viajo, etc.

DIAGRAMA DE GANTT



Proyecto: Proyecto Fecha: mar 01/10/13	Tarea		Resumen inactivo		Tareas externas	
	División		Tarea manual		Hito externo	
	Hito		solo duración		Fecha límite	
	Resumen		Informe de resumen manual		Progreso	
	Resumen del proyecto		Resumen manual		Progreso manual	
	Tarea inactiva		solo el comienzo			
	Hito inactivo		solo fin			

HERRAMIENTAS Y DISPOSITIVOS UTILIZADOS

Para la realización de World Of Wars se han utilizado hasta 5 dispositivos diferentes tanto para el desarrollo del juego, como para la fase de pruebas.

De estos cinco dispositivos, tres de ellos son ordenadores con diferentes sistemas operativos (Windows, Linux y Mac). Los otros dos dispositivos restantes son dos dispositivos Android, un Nexus 10, y un Smartphone LG Optimus 2X.

Las herramientas a utilizar han sido las siguientes:

- Visual Studio 2012 / GCC / XCode 4.0
- Git como sistema de control de versiones
- Adobe Photoshop CS3
- Microsoft Office para la generación de la documentación

USUARIOS Y CONTEXTO DE USO

En este apartado se va a detallar los procesos seguidos durante el desarrollo de la etapa de análisis en los usuarios y contexto de uso. Para ello se van a utilizar los métodos de indagación más oportunos para poder obtener toda la información sobre las necesidades, objetivos, y el contexto de los usuarios a los que la aplicación va dirigida. El conocimiento sobre los usuarios, sus contextos de uso, sus necesidades, objetivos así como también sus actitudes son imprescindibles para la realización de un buen diseño centrado en el usuario y para desarrollar aplicaciones y entornos usables.

Métodos de indagación

Justificación y planteamiento de los métodos escogidos

La aplicación a desarrollar, al ser un videojuego, va destinada a un público que aunque no es específico, si debe de cumplir dos requisitos mínimos y obligatorios:

- ✚ Ser usuario de un Smartphone
- ✚ Ser como mínimo un jugador casual de videojuegos

Aunque perfectamente se podría desarrollar un videojuego para aquellas personas que nunca han jugado a videojuegos, y podría ser un mundo muy grande en el que explorar, se ha impuesto el requisito de que al menos, a grandes rasgos, el público al que estará destinado la aplicación debe de disponer de un Smartphone así como también haber jugado alguna vez en su vida casualmente a videojuegos. Dicho esto, se procede a recopilar toda la información posible sobre las necesidades, objetivos, actitudes y contextos de uso de los usuarios utilizando dos métodos de indagación: La observación e investigación, y las encuestas.

Primeramente se han realizado las encuestas para así poder tener un primer filtro importante de usuarios que tal vez no interesan (usuarios sin Smartphone por ejemplo). La idea es realizar una encuesta inteligente para que aquellos usuarios que no interesen, automáticamente no sean contabilizados sus datos para no ensuciar así los datos de estas.

Seguidamente, aprovechando algunos de los usuarios de las encuestas, se ha procedido a realizar el método de observación e investigación a aquellos usuarios de los que se podría recopilar bastante información por su alta dedicación a los videojuegos.

Para la realización de la encuesta se ha procedido a utilizar los servicios de *Google Docs*, utilizando un formulario de *Google Drive* para que así la gente pudiera acceder fácilmente al cuestionario y disponer de todos los resultados de forma real, así como también unas estadísticas detalladas con mucha facilidad.

<https://docs.google.com/forms/d/170sFs2O9RWrgYBoZ1wONv1usbDpoS2CqJsYLrsimYB4/viewform>

Las preguntas que se han realizado han sido las siguientes:

Edad: Se ha considera importante conocer la edad, ya que como es evidente una persona con 60 años es posible que pocas veces haya jugado en su vida, sin embargo un joven de 16 años casi seguro que será bastante asiduo a los videojuegos.

Profesión o estudios: Aunque no sea realmente un dato importante, si ha sido útil conocer que profesión o estudios disponía de la persona.

Sistema Operativo Móvil: Una pregunta importante para ver una estadística de los diferentes sistemas operativos usados en los dispositivos móviles.

Tipo de jugador: Esta pregunta nos ayudará a saber cuál es el perfil del jugador que más se cumple en un entorno real.

Tiempo de dedicación a los juegos (General): Esta pregunta nos rebelará si los usuarios destinan mucho o poco tiempo a jugar a los videojuegos.

Tiempo de dedicación a los juegos Móvil: En este caso, podremos saber el porcentaje de juego que se destina a las plataformas móviles.

Tiempo medio de uso del móvil con juegos: Aunque esta pregunta, parezca parecida a la anterior, lo que realmente queremos saber es si realmente un usuario realiza partidas largas en el móvil en el día a día o partidas más cortas.

Juego competitivo o cooperativo: Con esta pregunta se pretende descubrir si la gente prefiere más los juegos competitivos versus los cooperativos.

Característica más valorada de un videojuego para móvil: Con esta selección se pretende averiguar que característica valoran más los usuarios para tenerla en cuenta.

Característica más atractiva de un videojuego para móvil: Esta pregunta se puede considerar una pregunta idéntica a la anterior, pero que nos desvelará posibles características extras que se le puedan añadir.

Género preferido para videojuego móvil: Aunque está pregunta está fuera del rango, puesto que el género del videojuego no se podría cambiar, si podría ayudar a adaptarlo más hacia un género que otro.

Ultimo videojuego jugado en el móvil: Por último se ha preguntado por el último videojuego jugado para tener una idea de lo que la gente suele jugar actualmente.

Resultados

De estas encuestas hemos podido obtener los siguientes resultados:

Edad:

La edad media entre los participantes oscila entre los 18-30 años. Es decir podemos considerar que la población que se ha realizado la encuesta es gente joven.

Sistema Operativo Móvil

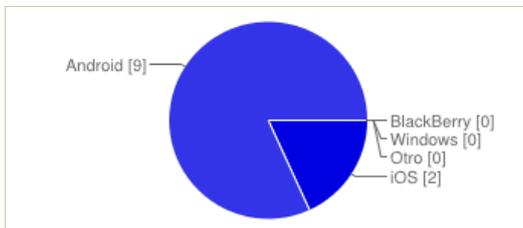


ILUSTRACIÓN 1: GRÁFICA DE OS MÓVIL

Se puede observar claramente que la gran mayoría de usuarios utilizan *Android*, aunque también una pequeña parte utiliza *iOS*. En un principio *Word of Wars* estará destinada a plataformas Android, pero es posible realizar la portabilidad a iOS con poco esfuerzo gracias a la tecnología usada.

Tipo de jugador

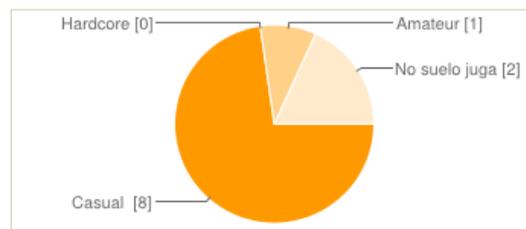


ILUSTRACIÓN 2: GRÁFICA DE TIPO DE JUGADOR

Esta información nos indica que la mayoría de jugadores son casuales. Así que nos tendremos que centrar sobretodo en este tipo de jugadores que serán los más frecuentes.

Tiempos

En el caso de los tiempos de dedicación a los juegos en general, existe una gran diversidad. Desde ninguna hora semanal, hasta las 20 horas. Así que de esta información poco podemos extraer. Lo que si podemos comprobar es que el tiempo de dedicación de los juegos en el móvil a la semana no supera las 4 horas en su gran mayoría (excepto en un caso puntual).

Respecto al tiempo medio que pasa la gente jugando a un videojuego suele ser de unos 20 minutos de media. Así que podemos corroborar que las partidas han de ser rápidas.

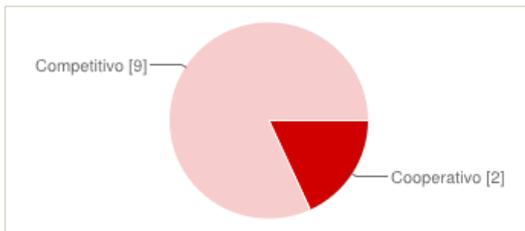


ILUSTRACIÓN 3: GRÁFICA DE COOPERATIVO VS COMPETIVO

Competitivo vs Cooperativo

En esta gráfica se puede observar que la gran mayoría de gente prefiere los videojuegos competitivos y enfrentarse en entre ellos en vez de cooperar.

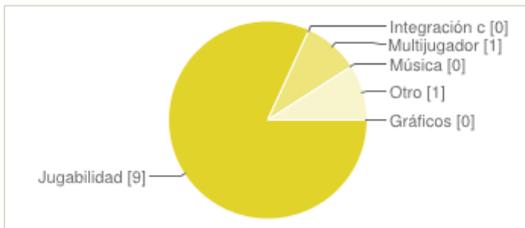


ILUSTRACIÓN 4: GRÁFICA DE CARACTERÍSTICAS MEJOR VALORADAS

Características

Finalmente, se ha llegado a la conclusión de que los jugadores prefieren los juegos con gran jugabilidad y que tengan una mecánica fácil siempre que se habla de juegos móvil.

En un principio parece que es la característica más importante. La segunda característica atractiva pero menos relevante es la de multi-jugador.

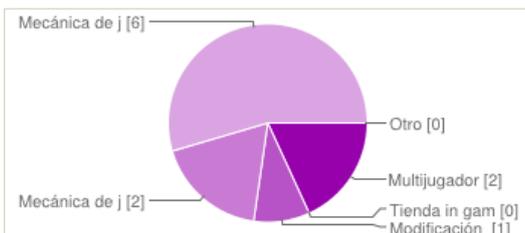


ILUSTRACIÓN 5: GRÁFICA DE CARACTERÍSTICAS MÁS ATRACTIVAS

Por último, vemos claramente los dos grandes géneros que se llevarían el pastel son los juegos casuales así como los de acción. Habrá que tomar nota de esto para poder realizar una mezcla de estos dos y de esta forma hacerlo aún más atractivo para todo el público.

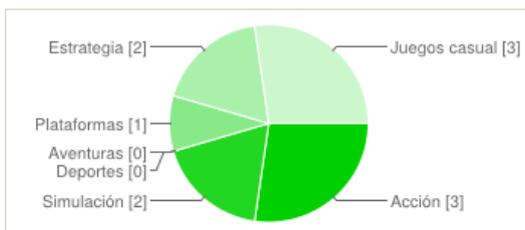


ILUSTRACIÓN 6: GRÁFICA DE GÉNEROS

Último juego jugado

Solo a modo de anécdota comentar que uno de los juegos más jugados por los encuestados ha sido *Candy Crush* (un videojuego con mecánica simple pero adictiva)

Para acabar la fase de investigación, se seleccionó a un antiguo compañero, que es un jugador asiduo, para observar e investigar si era posible extraer algún dato importante que se nos escapara con las encuestas. Se realizaron varias preguntas y comentarios respecto a posibles implementaciones que se podrían realizar al juego, y aunque algunas de ellas se habían planteado en un principio se descartaron por falta de tiempo. La conclusión final que se ha podido extraer de las observaciones son:

- ✚ La interfaz debe de ser cómoda para usar en un dispositivo táctil.
- ✚ Los controles deben de ser personalizables, ya sea usando la pantalla táctil, un joystick virtual, etc.
- ✚ La dificultad del juego debe de ser logarítmica para no asustar a los más novatos, pero hacer que los más expertos no se aburran muy pronto.

Perfil de usuarios identificados

De todos los datos obtenidos utilizando ambos métodos de indagación, se han podido extraer 3 tipos de usuarios diferentes. Como primera aproximación se podría decir que existen dos grupos muy diferenciados entre sí. Un grupo formado por la gente que suele ser jugar a los videojuegos aunque sea de forma casual, y el otro grupo de gente que casi nunca o por no decir nunca suele jugar a los videojuegos y mucho menos en plataformas móviles.

Los 3 perfiles de usuario para *World Of Wars* serán:

Usuario inexperto

Los usuarios inexpertos, son aquellos usuarios que no tienen casi experiencia alguna en el mundo de los videojuegos. Serán los usuarios que tal vez jueguen al videojuego por alguna recomendación de algún amigo o simplemente por el hecho de probar algo nuevo.

Usuario casual

Los usuarios casuales, son aquellos que suelen jugar a videojuegos, pero que no suelen dedicarle muchas horas y juegan de forma esporádica, sobre todo para pasar el rato cuando están en momentos de espera, por ejemplo en los viajes de metro, mientras espera a un compañero, etc. Estos jugadores desean tener partida rápidas en los momentos puntuales y de forma casual.

Usuario activo

Por último los usuarios activos, son aquellos usuarios que podríamos catalogar como usuarios más expertos en el mundo de los videojuegos y aquellos que les dedican bastante tiempo. Son usuarios muy competitivos y capaces de destinar bastantes horas en el videojuego y con ganas de superar records y ser el mejor. Estos jugadores, desean hacer todo lo posible para conseguir siempre ser los mejores y demostrar superioridad ante los demás.

Contexto de uso

World Of Wars al tratarse de un videojuego para dispositivos móviles dispone de un contexto de uso bastante amplio.

Hoy en día, cualquier usuario de un Smartphone puede jugar a *World Of Wars*, en cualquier momento del día, en cualquier sitio y siempre que lo desee. Así que sería interesante que el videojuego pueda ser adaptado para ser jugado en diferentes lugares, ya sea estando mientras el usuario está en la cama tumbado jugando un rato antes de acostarse, como si el usuario está jugando en el autobús mientras está de camino al trabajo. Tal y como se ha comentado anteriormente, dado que se puede jugar en cualquier lugar, también se puede jugar en cualquier momento del día. Ya sea durante el día o por la noche, durante períodos de tiempo no muy lagos que superen tal vez la hora de juego.

Respecto al dispositivo de uso, a ser un Smartphone es casi imprescindible disponer de conexión a internet así de esta forma se podrá tener sincronizado en todo momento las puntuaciones o todo el estado de juego en un servidor externo para poder continuar la partida más tarde o incluso en otros dispositivos.

Análisis de tareas

Las tareas mínimas que se desean implementar en *World Of Wars* son las siguientes:

Tarea 1: Iniciar una nueva partida

- a) Iniciar la aplicación
- b) Seleccionar “Nueva Partida”
- c) Seleccionar el jugador
- d) Introducir el nombre del jugador
- e) Iniciar la partida

Tarea 2: Continuar una partida anterior

- a) Iniciar la aplicación
- b) Seleccionar “Continuar Partida”
- c) Seleccionar la partida a cargar
- d) Iniciar la partida

Tarea 3: Cambiar el control del juego

- a) Iniciar la aplicación
- b) Seleccionar “Opciones”
- c) Cambiar el tipo de control del juego
- d) Volver al menú principal

Tarea 4: Acceder a la tienda de objetos

- a) Iniciar la Tarea 1 o 2
- b) En el menú del mapa, seleccionar la opción de “Tienda”

Tarea 5: Adquirir un nuevo objeto

- a) Iniciar la tarea 4
- b) Buscar el objeto que se desea adquirir siempre que se tenga suficiente dinero
- c) Comprar el objeto

Tarea 6: Equipar un objeto o modificar acciones:

- a) Iniciar la tarea 1 o 2
- b) En el menú del mapa, seleccionar la opción “Inventario”
- c) Seleccionar el objeto nuevo a equipar o a modificar
- d) Modificar las N acciones que desea disponer en el combate

DISSEÑO CONCEPTUAL

Después de recopilar toda la información necesaria en los apartados anteriores, procederemos a describir los escenarios de uso, así como también los flujos de interacción. Adicionalmente, teniendo en cuenta los perfiles de usuario detectados anteriormente, construiremos tres personas ficticias.

Personas

Se han creado tres personas ficticias, una por cada perfil de usuario detectado, para poder realizar los escenarios de uso.



ILUSTRACIÓN 7: MARÍA GONZÁLEZ

María González

28 Años | Profesora

“Nunca he jugado a videojuegos en los móviles”

Objetivos:

- Distraerse durante los viajes de metro o las esperas.
- Aprender a divertirse en el mundo de los videojuegos.

Comportamiento:

- Usaria de Smartphone para uso social.
- No aprovecha todo el uso que puede darle a un Smartphone.

Necesidades:

- Necesita un proceso de adaptación e instrucción previa.
- Interfaz simple e intuitiva.

Ilustración 1: By brianac37 [CC-BY-2.0 (<http://creativecommons.org/licenses/by/2.0/legalcode>)] vía Flickr

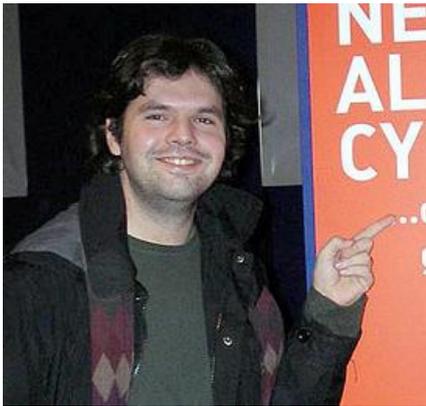


ILUSTRACIÓN 8: JUAN FERNÁNDEZ

Juan Fernández

29 Años | Administrador de sistemas

“Me encanta jugar para desconectar y pasar un buen rato.”

Objetivos:

- Desconectar del estrés de su día a día jugando.
- Aprovechar los tiempos muertos para avanzar un poco en el juego.

Comportamiento:

- Usuario videoconsolas y varios juegos de PC.
- Usuario con cierto nivel de experiencia y cierta exigencia en los videojuegos.

Necesidades:

- Necesita una lógica de juego que le enganche.
- El juego necesita de partidas no muy largas.

Ilustración 2: By hulshof [CC-BY-2.0 (<http://creativecommons.org/licenses/by/2.0/legalcode>)] vía Flickr



ILUSTRACIÓN 9: MIGUEL JACOBO

Miguel Jacobo

23 Años | Estudiante de ingeniería

“Quiero ser el número uno y que no haya rival alguno para mí”

Objetivos:

- Ser el mejor y ganar a todos sus compañeros.
- Conseguir los logros más difíciles y reventar las puntuaciones.

Comportamiento:

- Usuario experto en muchos videojuegos.
- Usuario con mucho tiempo libre para dedicarle a los videojuegos.

Necesidades:

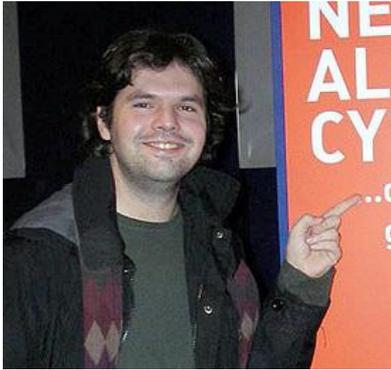
- Un sistema de dificultad adaptable para aquellos jugadores más profesionales.
- Estas dificultades más extremas requieren más tiempo así como ciertas habilidades para poder superar los niveles.

Ilustración 2: By 20after4 [CC-BY-2.0 (<http://creativecommons.org/licenses/by/2.0/legalcode>)] vía Flickr

Escenarios de uso



María se encuentra con todos sus amigos mientras están probando un juego nuevo que acaba de salir. Le recomiendan que se instale la aplicación y la prueba. María se lo instala y al ser un juego intuitivo y bastante fácil en un principio María se divierte y comenta con sus amigos.



Juan, se encuentra un día aburrido después de trabajar y decide mirar en la Play Store si existe algún juego que le ayude a quitar ese aburrimiento mientras viaja en el metro hacia casa. Juan descubre *World Of Wars* y decide instalarlo. Empieza a jugar y sin darse cuenta ha pasado más de 45 minutos jugando y divirtiéndose y está a punto de llegar a su destino.



Miguel se encuentra leyendo un día un blog sobre últimas tendencias de videojuegos y encuentra que *World Of Wars* es la última moda entre los jóvenes. Así que decide instalárselo y empezar una nueva partida. Como se considera un jugador de los buenos, intenta conseguir los máximos logros posibles y no realizar ningún fallo. Así que se pasa jugando durante más de 2 horas hasta que agota la batería de su *tableta*.



Los amigos de María crean un grupo de *Whatsapp* para así poder ir comentando como van avanzando con su videojuego. María se encuentra atascada porque se le hace más difícil el juego. Hasta que se da cuenta de que tiene el dinero suficiente para poder actualizar las armas para así poder disparar con más fuerza y eliminar antes a los enemigos y poder avanzar de nivel



Miguel se ha pasado varios niveles consiguiendo todos los objetos y logros. De esta forma ha accedido a unos niveles secretos. Antes de acceder a los niveles, decide ir a su inventario para reparar su avión y tenerlo equipado al máximo posible para enfrentarse a los posibles súper-enemigos.

Flujos de interacción

El flujo de interacción nos permite representar de forma visual las decisiones que tomarán los usuarios para llegar al objetivo deseado. Podemos observar que se diferencian en dos colores. Los que son de un tono amarillento son los flujos de interacción por los menús principales. Los que son en tonos azulados corresponden al flujo de interacción una vez ya está la partida de juego iniciada.

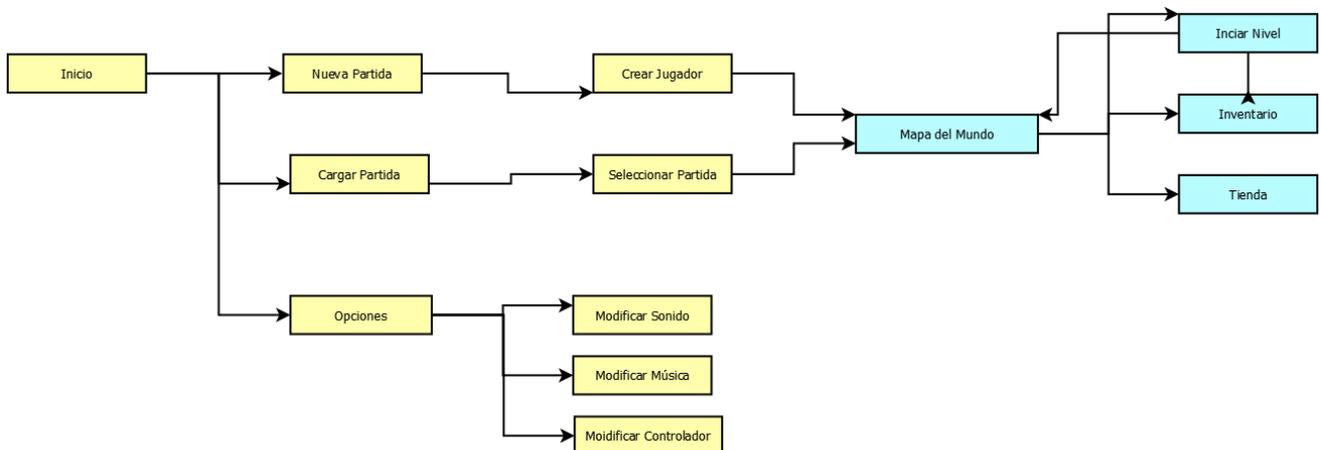


ILUSTRACIÓN 10: FLUJO DE INTERACCIÓN

PROTOTIPAJE

Sketches

Después de la obtención de los resultados del análisis competitivo y la observación contextual, se han llegado a elaborar los primeros Sketches a mano alzada del juego *World Of Wars*.

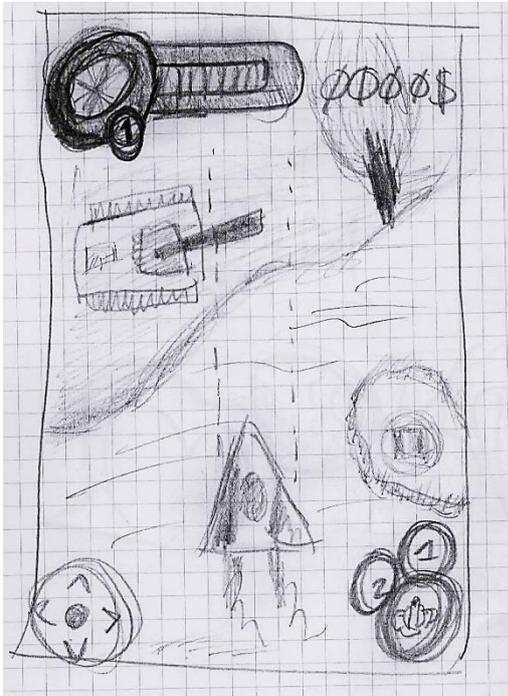


ILUSTRACIÓN 11: SKETCH GAME-PLAY

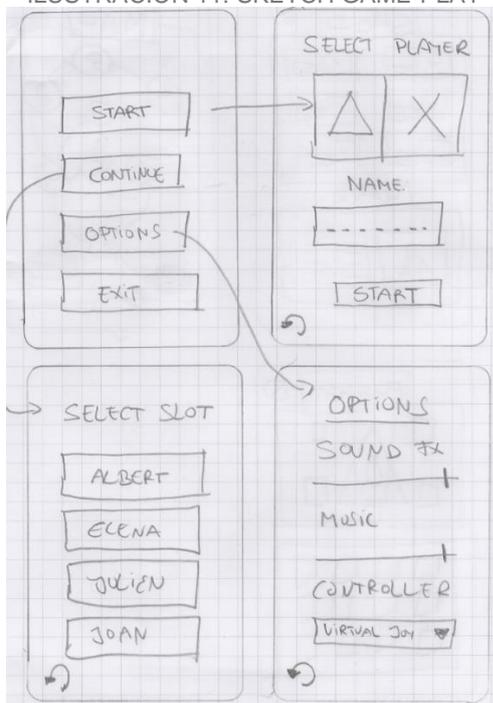


ILUSTRACIÓN 12: SKETCH MENÚS 1

En este sketch, podemos observar una primera impresión de cómo será la pantalla del juego cuando ya está en progreso. Se ha decidido de desarrollar primeramente esta pantalla, ya que es donde transcurre la acción del juego y la cual estará activa más del 80% del tiempo.

En estos sketches, podemos observar las pantallas de menú iniciales así, como también el flujo entre ellas.



ILUSTRACIÓN 13: SKETCH MENÚS 2

Por último, se muestra los sketches de tres partes bastante importantes del juego que son. La tienda de mejora de objetos, el mapa global para poder interactuar con los diferentes niveles y escenarios, y también la pantalla de inventario para poder equipar o cambiar los objetos, así también como las facultades del jugador.

Prototipo de alta fidelidad

Gracias los primeros bocetos creados anteriormente, es posible disponer de una idea global que como será la interfaz a modo general. Seguidamente se mostrará un prototipo de la interfaz más detallada y con algunas explicaciones breves.

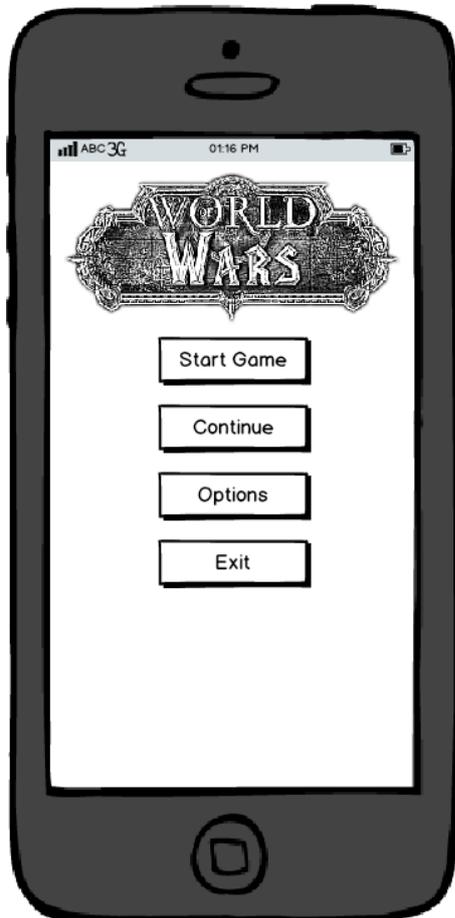


ILUSTRACIÓN 14: PROTOTIPO PANTALLA INICIAL

INICIO

Esta es la pantalla principal que es visible una vez se ha ejecutado el juego. Desde esta pantalla disponemos del menú principal para poder empezar una nueva partida, continuar una partida que ya ha sido empezada en un momento anterior, modificar las opciones del juego, y por último salir del juego.

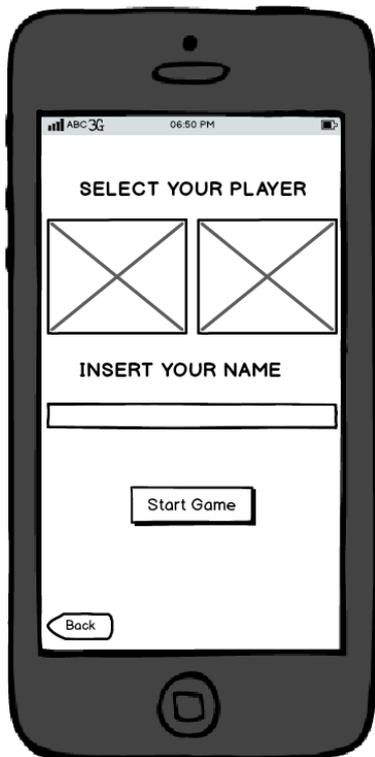


ILUSTRACIÓN 15: PROTOTIPO NUEVA PARTIDA

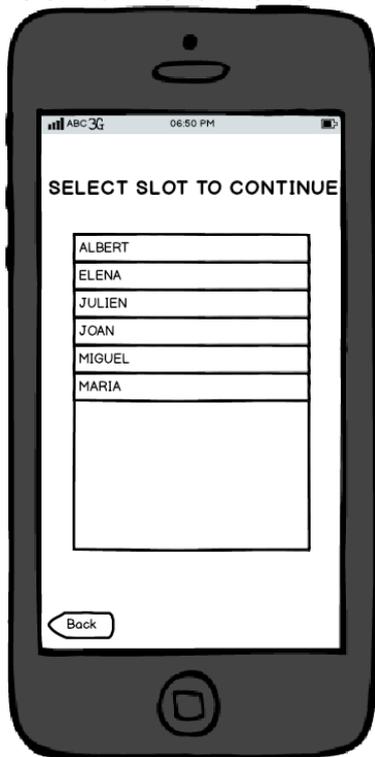


ILUSTRACIÓN 16: PROTOTIPO CONTINUAR PARTIDA

NUEVA PARTIDA

Desde aquí el jugador podrá seleccionar el avión que desea controlar, e insertar su nombre para en un futuro saber cuál es la partida que debe de continuar.

CONTINUAR PARTIDA

Una vez un jugador haya creado la partida desde este apartado será posible volver a continuar la partida en el último nivel donde lo dejó el jugador.

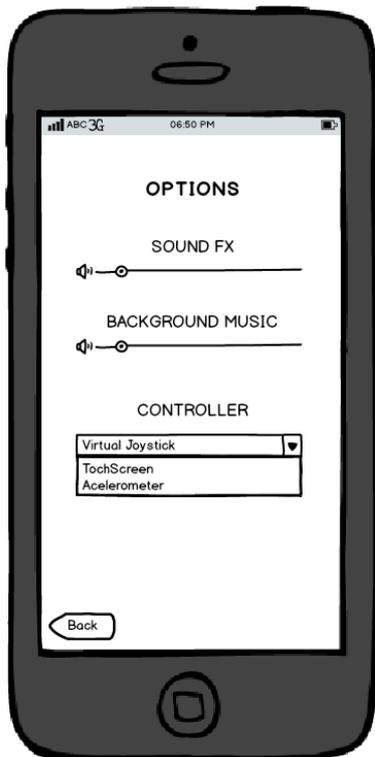


ILUSTRACIÓN 17: PROTOTIPO OPCIONES

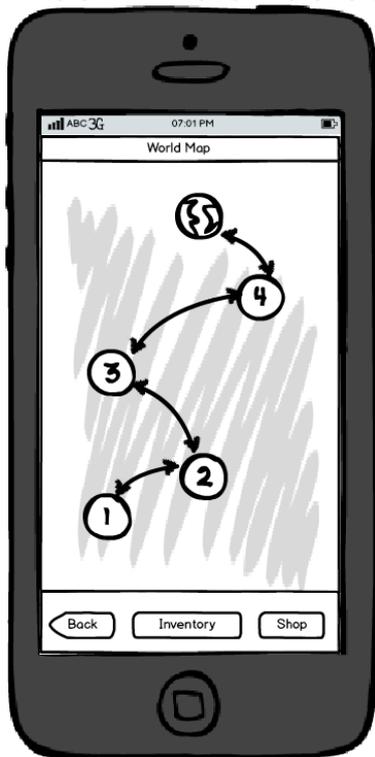


ILUSTRACIÓN 18: PROTOTIPO MAPA

OPCIONES

Desde la pantalla de opciones se podrá controlar los diferentes volúmenes tanto de los efectos especiales como de la música. También será posible modificar el tipo de control que se desea para el juego, ya sea utilizando un joystick virtual, tocando directamente la pantalla o bien usando el acelerómetro si está disponible.

MAPA DEL MUNDO

Una vez se haya arrancado el juego y estemos dentro de una partida, desde esta interfaz podremos seleccionar el nivel que queremos jugar, o bien también podemos acceder desde aquí a la tienda para comprar nuevas mejoras para nuestro avión. También es posible acceder a nuestro inventario para realizar el cambio de estas nuevas mejoras.

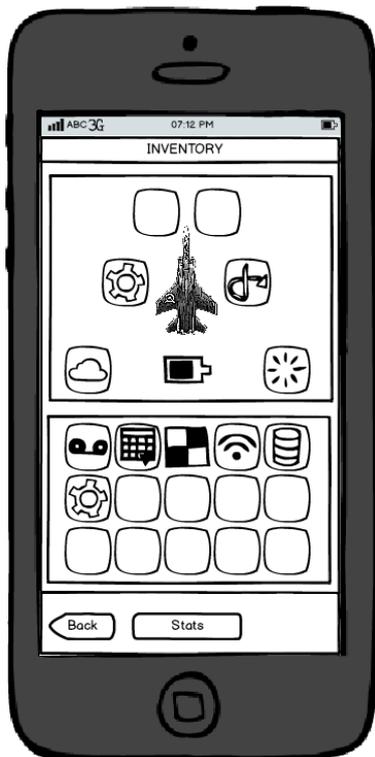


ILUSTRACIÓN 19: PROTOTIPO INVENTARIO

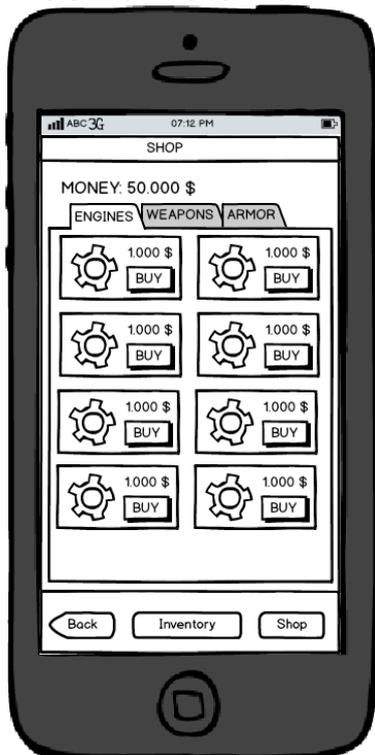


ILUSTRACIÓN 20: PROTOTIPO TIENDA

INVENTARIO

Desde el inventario podremos realizar tanto los cambios de las mejoras de nuestro avión así como de las diferentes habilidades, y poder ver las estadísticas generales de nuestro avión.

TIENDA

Desde la tienda se podrá adquirir nuevas mejoras, habilidades, municiones extra, etc. Podremos ver el dinero disponible y el precio de cada pieza y la opción de comprarla. Desde esta pantalla será posible también seguidamente acceder al inventario para poder realizar los cambios pertinentes.

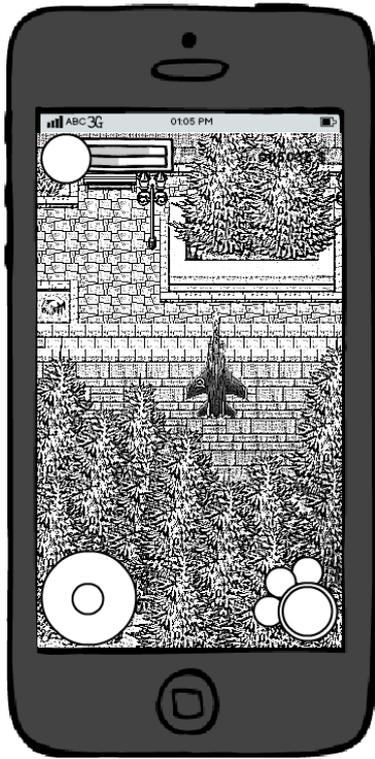


ILUSTRACIÓN 21: PROTITPO GAME-PLAY

DENTRO JUEGO

La pantalla más importante es la que ocurre una vez el usuario este en pleno juego, ya que es donde gastará la mayor parte del tiempo. Desde aquí se pueden observar los diferentes controles como el caso del joystick virtual en el caso de lo que tenga activado así como los botones inferiores de la derecha para activar las habilidades.

En la parte superior se mostrará la información del jugador tal como su nivel, la vida del avión, así como también estadísticas como el dinero acumulado.

EVALUACIÓN

Test de usuarios

En el proceso de evaluación, se preparará una lista de tareas a realizar por los diferentes usuarios. La idea es comprobar si la usabilidad del prototipo así como de los procesos es la correcta. Para poder realizar el proceso de evaluación será necesario realizar los test en el contexto. Para ello, se deberá instalar el prototipo en su dispositivo móvil. El prototipo guardará un pequeño 'log' con los tiempos de ejecución y la situación geográfica vía GPS para saber si el usuario está en movimiento o bien está parado. La idea es poder extraer la información principal de cuando ha estado realizando las pruebas, ya que es de vital importancia conocer el contexto en el que estaba el usuario.

Las tareas a realizar por el usuario serán:

- Empezar una nueva partida
- Intentar superar el primer nivel
- Una vez superado salir del juego
- Continuar la partida que se ha creado anteriormente
- Adquirir una nueva mejora desde la tienda
- Instalar la mejora desde el inventario

Finalmente, el usuario deberá de responder un breve cuestionario para saber su impresión.

TEST DE USUARIO

Pregunta	Si	No	Ns/Nc	Comentarios
¿Has podido realizar todas las tareas?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
¿Ha sido fácil el control del jugador?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
¿Has intentado cambiar el control del jugador?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
¿Has tenido algún problema en entender cómo controlar a tu jugador?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
¿Has tenido algún problema al cargar tu partida anterior?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
¿Has tenido algún problema para adquirir tu mejora desde la tienda?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
¿Has tenido algún problema para equipar/instalar la mejora adquirida?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
¿Crees que el juego es intuitivo?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
¿Crees que se podría mejorar algún aspecto?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	

IMPLEMENTACIÓN

Para la implementación de World Of Wars, tal y como se definió en el documento de análisis de requisitos, se utilizara *framework* multiplataforma Cocos2D-X. Está basado en cocos2d-iphone, pero en vez de utilizar Objective-C, está programado en C++. Funciona en iOS, Android, OSx, Windows, Linux, Emscripten, Google Native Client, BlackBerry Tizen. Al ser un Framework bastante generalista, se ha intentado adaptar el *framework* para nuestras necesidades específicas.

ARQUITECTURA

En un principio la idea principal era utilizar el patrón de MVC (Modelo Vista Controlador) para la creación del motor principal del juego y poder dividir tanto la parte del modelo, como la parte lógica así como también la parte gráfica. El problema radica en el framework que hemos utilizado.

Las clases de Cocos2dx (*CCSprite*, *CCLayer*, *CCScene*) son subclases de *CCNode*.

La implementación más simple de un juego en Cocos2DX sigue los siguientes procesos que operan entre ellos para implementar la lógica del juego.

- ✚ Se crea una instancia de una *CCScene*.
- ✚ *CCScene* instancia una o más *CCLayers* y la añade como hijos.
- ✚ *CCLayer* instancia uno o más *CCSprites* y los añade como hijos
- ✚ *CCScene* se encarga de la entrada del usuario (tales como eventos táctiles, o cambios en el acelerómetro), o ejecuta diferentes *CCAction* para manejar los comportamientos de los objetos.
- ✚ *CCScene* ejecuta un bucle para actualizar los *CCLayers/CCSprites* en intervalos de tiempo.

El problema de esto es que desde este enfoque una *CCScene* era la responsable de controlar muchas tareas, como la interacción del usuario, la lógica del juego así como también la visualización.

Por ello, en vez de usar directamente las clases de Cocos2dx, se ha preferido dividir cada parte importante en clases y subclases y desarrollar un sistema de control para estas clases e intentar acercarse al patrón MVC.

La clase más importante que se utiliza la programación del juego es la entidad básica llamada *Entity*. Esta es la responsable de controlar la parte gráfica y visible sin controlar ninguna lógica. La lógica de cada elemento del juego será controlada y dispondrá de una lógica u otra gracias a la herencia múltiple y al poliformismo. De esta forma por ejemplo, las *Player* y *Enemy*, son dos objetos *Entity* pero que reaccionan de forma diferente en su llamada de *Update*.

Estas entidades son controladas por unos managers. En el caso de las *Entity*, el encargado de controlarlas es la clase *Map*. Es la contenedora donde se insertan las entidades con las que se puede interactuar en el juego.

La transición de las pantallas en todo momento está controlada por *ScreenManager* que es la encargada de pasar de una pantalla a otra, e incluso de actuar como una pila (*stack*) para poder sacar (*pop*) o empujar (*push*) pantallas sin tener que definir un flujo.

En todo momento existe una gran clase que controla los aspectos más globales del juego que es *GameManager*, tales como el control de carga o guardado de una partida, pausar o reanudar el juego, etc.

Como último manager, disponemos del *ImprovementManager*, que es el encargado de las mejoras para nuestro jugador. Es el encargado inicializar las mejoras y mantenerlas sincronizadas con el jugador actual, así como la tienda de objetos.

Estos managers utilizan el patrón *Singleton* para poder ser llamados y usados desde cualquier parte del código y asegurarnos que solo existe una instancia de este.

Patrón Singleton

```
template<typename T>
class Singleton
{
private:
    static T *m_pInstance;

public:
    static T* Instance()
    {
        if (!m_pInstance)
            m_pInstance = new T();
        return m_pInstance;
    }
};
```

El funcionamiento básico de la lógica de ataque es el siguiente:

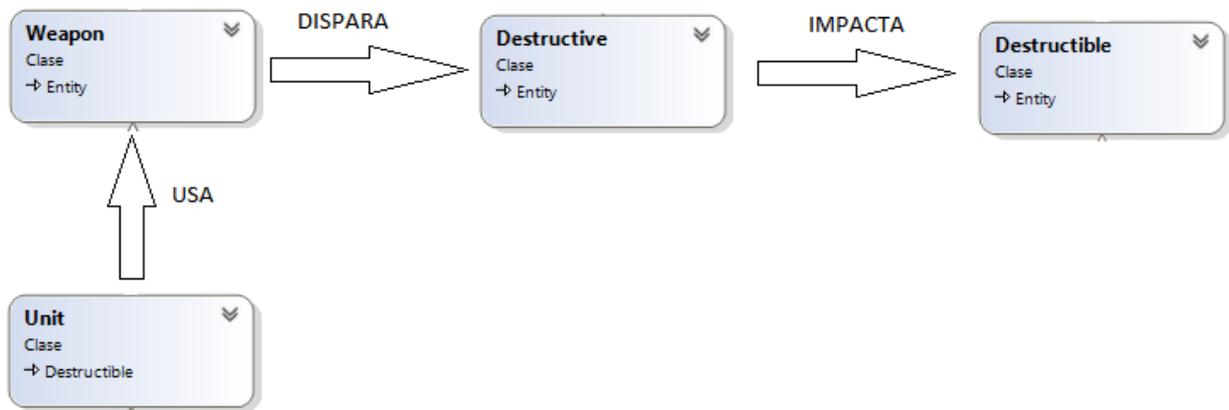


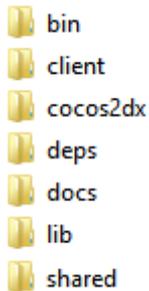
ILUSTRACIÓN 22 DIAGRAMA DE LA LÓGICA DE ATAQUE

Una clase *Unit*, dispone de una o varias armas (*Weapons*). Estas armas son capaces de disparar proyectiles, llamados *Destructive*. Estos proyectiles salen lanzados desde el punto de origen donde está el arma, así como también con su rotación actual y siguiendo el vector normal. Los proyectiles impactan con cualquier objeto de la clase *Destructible* (o bien que herede de él). Una vez ha impactado generará un daño según el arma que lo ha lanzado.

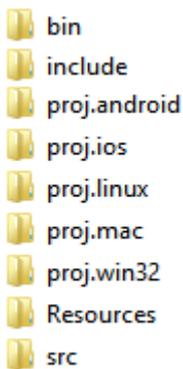
En todo momento se ha intentado tener la referencia de quienes son las fuentes (*source*) que han generado los eventos, y quienes son los objetivos (*target*) para de esta forma poder realizar una lógica u otra.

REFERENCIA DE ARCHIVOS

Desde este apartado se hará un breve repaso de los archivos fuentes más importantes así como sus clases que se han desarrollado. Antes de todo se explicará un poco la estructura de los directorios.



En la raíz principal podemos observar la estructura principal del proyecto. Las partes realmente importantes son los directorios del cliente, el directorio del framework a utilizar (*cocos2dx*) y también el directorio *shared*. En este último directorio llamado *shared* se encuentra una pequeña librería que se ha desarrollado para que fuera común para otros futuros proyectos o incluso para la implementación de la parte multijugador que se podrá realizar en un futuro. La carpeta que más nos interesará es la del cliente del juego en sí.



En esta carpeta podemos observar que existe una carpeta para cada proyecto según el entorno a compilarlo. Por ejemplo en el caso de compilar el proyecto para Android se deberá de acceder al directorio *proj.android*. El código del proyecto está dividido en los ficheros de código fuente .CPP localizados en el directorio *src* y los encabezados (.h) en el directorio *include*. También se puede comprobar la carpeta *Resources* que es donde se incluyen todos los recursos gráficos, sonidos, etc. necesarios para el proyecto.

El código fuente se ha dividido en varios grupos para así tener el código más organizado y limpio. Aunque existen diferentes grupos, básicamente se pueden diferenciar cuatro grandes grupos:

-  Utils
-  Managers
-  Entities
-  Screen

UTILS

En el código fuente de este directorio se encontrarán diferentes utilidades que nos ayudarán a ciertas tareas o a extender nuevas funcionalidades. Existen varias nuevas funcionalidades extras que se han añadido a Cocos2DX como es el caso de *ParallaxScrollNode* o el *LayerPanZoom*. Las clases que se han implementado han sido las siguientes:



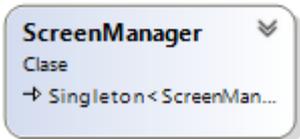
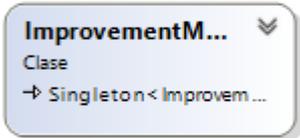
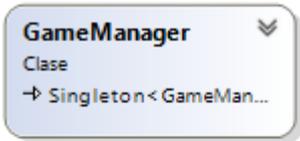
ILUSTRACIÓN 23 DIAGRAMA DE 'UTILS'

Touchable: Es una clase encargada de recibir los eventos táctiles realizados por el usuario. Cualquier clase que herede de esta podrá recibir los eventos táctiles.

Joystick: Es una clase encargada de gestionar y mostrar un joystick virtual en pantalla. De esta forma se podrá controlar el videojuego con este joystick en pantalla.

MANAGERS

Los managers en un principio todos heredan del patrón *singleton* permitiendo así existir tan solo una instancia de dicho manager y accesible desde cualquier parte del código. Los dos más importantes son *GameManager* y *ScreenManager*.



GameManager: Es el encargado de gestionar el juego en sí. Desde aquí se puede acceder al jugador actual, iniciar una nueva misión, acceder a las variables de la configuración de sonido, etc.

ScreenManager: Es el encargado de gestionar todas las pantallas del juego. Desde este manager se podrán crear nuevas pantallas, guardar la pantalla actual en una pila (*push*) o sacar la pantalla actual de la pila (*pop*). Este manager también nos permite realizar la transición entre las pantallas.

ImprovementManager: Es el encargado de gestionar las mejoras de las unidades. Se encarga de cargar las nuevas mejoras y de la gestión de estas.

ILUSTRACIÓN 24 DIAGRAMA DE LOS MANAGERS

ENTITIES

Las entidades son clases que se utilizarán para gestión de forma más óptima y eficiente y son las siguientes:

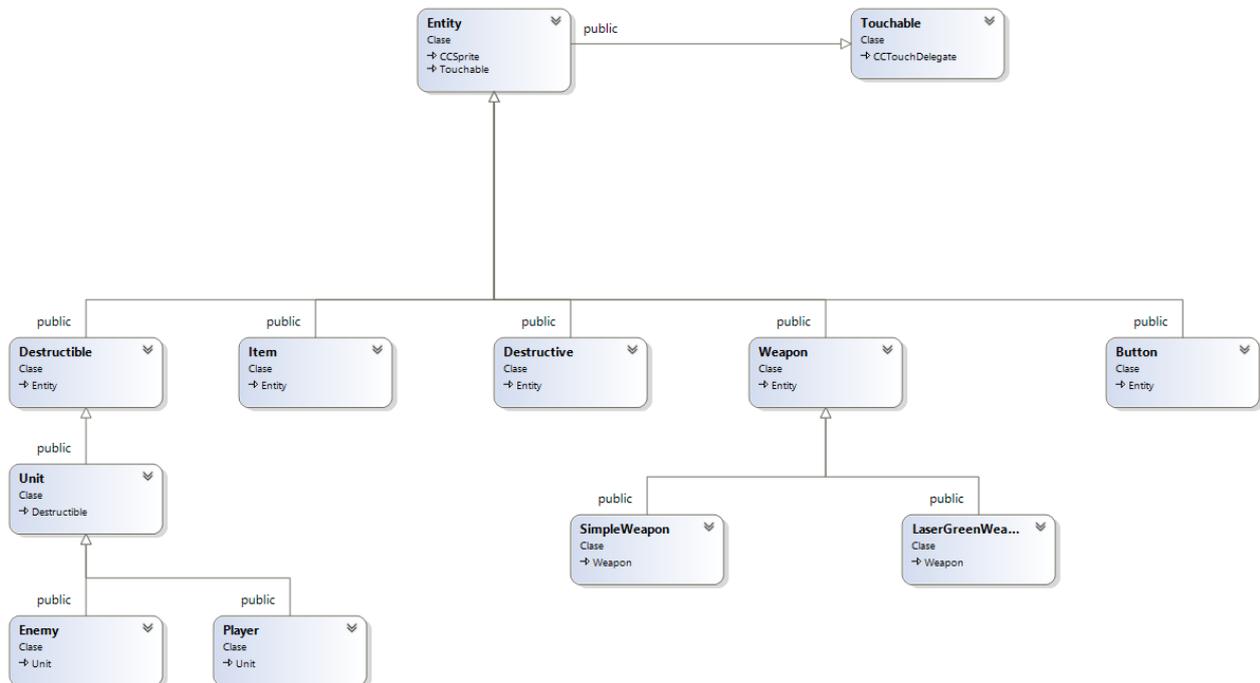


ILUSTRACIÓN 25 DIAGRAMA DE LAS ENTIDADES

Entity: es la clase base que utilizan la gran mayoría de las otras entidades. Está hereda de *CCSprite* proveniente de Cocos2Dx. Gracias a esta entidad nos abstrae un poco de la forma de mover un *sprite* gráfico por la pantalla, así como también controlar sus colisiones. Esta clase, también hereda de *Touchable*, lo que nos permitirá recibir los eventos táctiles para poder interaccionar con ella.

Button: Clase que hereda de *Entity* utilizada para utilizar elementos como botones y realizar ciertas llamadas cuando sean pulsados.

Item: Clase que hereda de *Entity* utilizada para crear objetos que son lanzados por los enemigos al morir y que pueden ser recogidos por el jugador.

Destructive: Clase base que hereda de *Entity* utilizada como elemento destructivo. Se podría decir que los *Destructive* son aquellos elementos que hacen daño a una clase de tipo *Destructible*. Un ejemplo de estos son las balas lanzadas por las armas.

Weapon: Clase que hereda de *Entity* también que se utiliza como armas en el juego. Es la encargada de lanzar elementos *Destructive*. En todo momento se encarga de controlar donde están los *destructives* lanzados y las colisiones entre ellos.

Destructible: Clase que hereda de *Entity* y se puede destruir. Un *Destructible* es todo aquel elemento del juego que puedo recibir daño (en este caso que colisiona con los *Destructive*).

Unit: Clase que hereda de *Destructible* y que funciona como una unidad de juego. Este tipo de entidad dispone de ciertas estadísticas tales como la vida, escudos, etc. También pueden equipar armas para poder atacar.

Enemy: Clase que hereda de *Unit*, la cual dispone de una inteligencia artificial usando el *BaseIA* para intentar atacar al jugador principal.

Player: Clase que hereda de *Unit*, la cual es controlada por el jugador. Esta clase única en todo el juego.

SCREEN

La clase *Screen*, es una clase base contenedora de diferentes objetos. En ella se añaden básicamente objetos de tipo *Entity* así como cualquier objeto que herede de la clase *CCNode* de Cocos2dx. Cada *Screen* se puede considerar como una pantalla diferente del juego, así de esta forma queda el código más localizado y organizado.

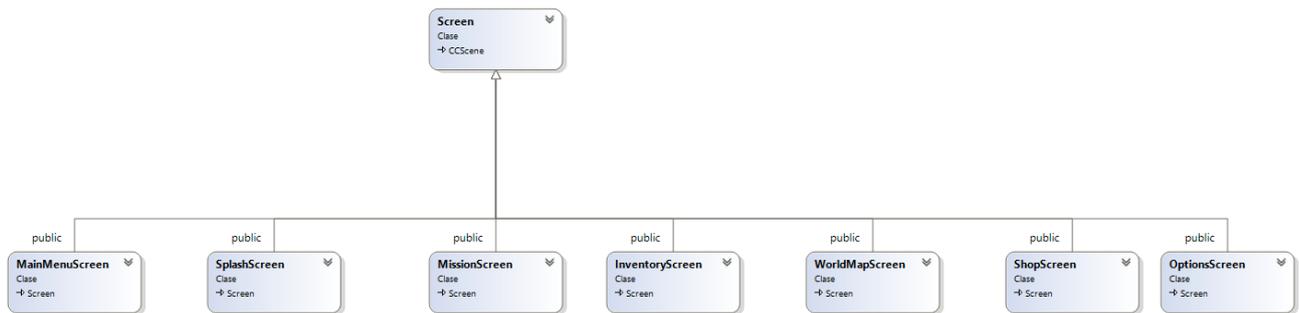


ILUSTRACIÓN 26 DIAGRAMA DE LAS PANTALLAS



ILUSTRACIÓN 27 CAPTURA DEL MENU



ILUSTRACIÓN 28 CAPTURA DE LAS OPCIONES

MainMenuScreen

Pantalla del menú donde podremos escoger iniciar un nuevo juego, continuar una partida o bien entrar a modificar las opciones del juego.

OptionsScreen

Pantalla donde podremos modificar las opciones del juego, tales como los volúmenes de sonido.



ILUSTRACIÓN 29 CAPTURA DEL MAPA

WorldMapScreen

Pantalla del mundo donde podremos movernos por las diferentes misiones del juego. Además desde esta pantalla se podrá acceder a la tienda de mejora de objetos.



ILUSTRACIÓN 30 CAPTURA DE LA TIENDA

ShopScreen

Pantalla donde se podrán adquirir nuevos objetos y mejoras para el jugador principal.



ILUSTRACIÓN 31 CAPTURA DEL JUEGO EN UNA MISIÓN

MissionScreen

Pantalla donde transcurrirá la mayoría del tiempo y donde se desarrolla la acción del juego. En esta pantalla se puede observar que existe un gran marco donde ocurre la acción, y otro marco inferior con los controles táctiles del jugador simulando un joystick virtual.

CONCLUSIONES

A modo general, estoy bastante satisfecho y contento con el resultado del proyecto. Es cierto que realizar un videojuego no es una tarea sencilla y realizarlo como proyecto de final de carrera con la limitación de tiempo del que disponíamos, ha sido todo un logro.

Aunque realmente conozco bastante el mundo de desarrollo de los videojuegos, nunca había podido dedicarle el tiempo necesario para poder crear uno desde el principio hasta el final, y no dejarlo en un simple concepto de videojuego o demo.

Me hubiera gustado haberle podido dedicar mucho tiempo y poder haber sacado una versión algo más divertida y elaborada de nivel de jugabilidad. Aunque a nivel personal intentaré seguir el proyecto hasta que sea digno de ser publicado en la Google Play Store.

Gracias al proyecto, he podido aprender bastante sobre el *framework* Cocos2DX con el que he adquirido cierta agilidad para la hora de desarrollar nuevos proyectos personales.

Por último me gustaría agradecer a todo aquel que ha hecho posible que este proyecto haya llegado a su fin:

- A mis tutores del proyecto: *Marc Domingo Prieto* y *Jordi Almirall López*
- A mis compañeros por darme ideas y sugerencias: *Aritz, Miguel, Pau*
- A mi novia *Elena* por apoyarme y ayudarme en los momentos más difíciles y estresantes.
- A todos aquellos que han puesto su granito de arena.

FUENTES DE INFORMACIÓN

Websites

- ✚ Cocos2DX API Reference
<http://www.cocos2d-x.org/wiki/Reference>
- ✚ Cocos2DX Forums
<http://www.cocos2d-x.org/forums>
- ✚ GameDev StackExchange
<http://gamedev.stackexchange.com/>
- ✚ OpenGameArt
<http://opengameart.org/>
- ✚ Parallaxer Game Starter Kit
<http://parallaxer.com/cocos2d-x-book/>

Libros

- ✚ Cocos2d-X by Example Beginner's Guide (Packt Publishing) by Roger Engelbert
<http://www.packtpub.com/cocos2d-x-by-example/book>