

A.Invasion

Ferran Mateos Herrero
Grado en ingeniería informática

Helena Boltà Torrell

Antonio Rodríguez Gutiérrez

08/01/2016



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FITXA DEL TREBALL FINAL

| | |
|--|---|
| Títol del treball: | <i>A. Invasión</i> |
| Nom de l'autor: | <i>Ferran Mateos Herrero</i> |
| Nom del consultor: | <i>Helena Boltà Torrell y Antonio Rodríguez Gutiérrez</i> |
| Data de lliurament (mm/aaaa): | <i>01/2016</i> |
| Àrea del Treball Final: | <i>Desarrollo aplicaciones android</i> |
| Titulació: | <i>Grado Ingeniería informática</i> |
| Resum del Treball (màxim 250 paraules): | |
| <p>El presente trabajo de final de grado consiste en un proyecto de desarrollo de un juego para dispositivos móviles con sistema operativo Android, abarcando todas las fases de un proyecto de desarrollo de software; des de la planificación hasta la obtención de un producto final.</p> <p>El juego desarrollado consiste en una versión básica de un juego de estrategia en tiempo real y pertenece al ámbito de juegos que se conocen como incrementales o numéricos al estar basado en cálculos matemáticos, que pueden ser alterados indirectamente por el usuario a través de un conjunto de acciones limitadas que se le presenta; mejorar habilidades principalmente.</p> <p>El principal objetivo del juego consiste en llevar a una raza alienígena invasora a extinguir toda la población humana, para dotar de un poco más de realismo al juego se han empleado datos y modelos matemáticos cercanos a la realidad siempre que ha sido posible.</p> <p>Respecto al proyecto se ha llevado a través de un modelo de gestión de proyectos tradicional e iterativo usando metodología de DCU para la obtención de requisitos y evaluación, especialmente de la interfaz.</p> <p>Finalmente la implementación se ha llevado a cabo mediante Android Studio, siguiendo el <i>framework</i> con un patrón MVC. Esta implementación se ha llevado a cabo des de cero y sin emplear librerías externas. Las pruebas se han realizado en emulador; nexus 5,4 y 9, y en dos dispositivos físicos; Nexus 5 y Samsung Galaxy s4.</p> | |

Abstract (in English, 250 words or less):

The current final degree work consists on a development Project of a game for Android based mobile devices, covering all phases of a software development Project; from the design to the end product.

The game consists on a basic version of a strategy game belonging to the incremental or mathematical game style as it is based on a series of formulas and calculations, which can be altered indirectly by user's limited actions; mainly skills upgrade.

The main objective in the game consist on getting an alien race to eradicate human population, to achieve a higher reality level real data and mathematical models have been used as much as possible.

In relation to the project itself, it has been carried out following a traditional project management approach with an iterative model. DCU methodology has been used in order to obtain requirements, with a special relevance on UI design.

Finally, the implementation has been performed with Android Studio, following main Android framework guidance and a MVC pattern. The game has been developed from scratch and without third party libraries. The testing has been conducted with emulators; using Nexus 5, 4 and 9 profiles, and also on two physical devices; Nexus 5 and a Samsung Galaxy S4.

Paraules clau (entre 4 i 8):

Android, A.Invasion, juego de estrategia, desarrollo app

Índice

| | |
|--|----|
| 1. Introducción..... | 3 |
| 1.1 Contexto y justificación del trabajo | 3 |
| 1.2 Objetivos del trabajo..... | 3 |
| 1.3 Enfoque y método seguido..... | 4 |
| 1.4 Planificación del Trabajo | 6 |
| 1.5 Breve descripción de los productos obtenidos | 7 |
| 1.6 Breve descripción del resto de capítulos de la memoria | 7 |
| 2. El Juego | 9 |
| 2.1 Objeto del Juego | 9 |
| 2.2 Skills..... | 9 |
| 2.3 Puntos de evolución | 10 |
| 2.4 El mapa | 10 |
| 2.5 Movimiento | 11 |
| 2.6 Crecimiento | 12 |
| 2.7 Muerte y condición de victoria | 12 |
| 2.8 Extensión..... | 13 |
| 3. Planificación | 14 |
| 3.1. Encuesta a usuarios | 14 |
| 3.2. Benchmarking | 16 |
| 3.3. Escenarios de uso | 17 |
| 3.4. Perfiles de usuario..... | 19 |
| 3.5. Conclusiones..... | 20 |
| 3.5. Casos de uso y flujos de interacción | 21 |
| 4. Diseño | 24 |
| 4.1 UI..... | 24 |
| 4.2 Componentes | 26 |
| 4.3. Arquitectura y diseño..... | 27 |
| 5. Implementación..... | 30 |

| | |
|-----------------------|--------------------------------------|
| 5.1 SurfaceView | 30 |
| 6. Conclusiones..... | 32 |
| 4. Glosario | 34 |
| 5. Bibliografía..... | 35 |
| 6. Anexos | ¡Error! Marcador no definido. |

1. Introducción

1.1 Contexto y justificación del trabajo

La temática escogida para el este Trabajo de final de grado consiste en un proyecto completo de desarrollo, incluyendo todas las fases, de un nuevo juego para dispositivos móviles con SO Android, al que hemos llamado “A. Invasión”.

El presente proyecto se enmarca dentro de los estudios oficiales del Grado de informática especialidad con especialidad en desarrollo de *software* por lo que se ha tenido en especial consideración a lo largo de todo el trabajo las fases típicas de la gestión de proyectos, ya que se entiende que más allá del producto final obtenido es de especial importancia aplicar las habilidades obtenidas a lo largo de la carrera, tanto las habilidades típicas de la arquitectura de programario como las transversales.

Teniendo en cuenta lo anteriormente expuesto así como que la aplicación desarrollada consiste en un juego se ha considerado que la finalidad básica del proyecto es adquirir nuevos conocimientos sobre el desarrollo de aplicaciones para dispositivos Android así como poner en práctica el conocimiento teórico ya adquiridos. No debemos olvidar que el producto final es un juego y por lo tanto se entiende que este debe satisfacer también la necesidad de entretenimiento de los usuarios finales, en concreto para “A. Invasión” se decidió enmarcar el juego dentro de la oferta de juegos de estrategia que existen actualmente y no se pretende en ningún momento realizar un juego educativo. Aún sin pretender ser un juego educativo, se ha intentado emplear siempre datos, formulas y planteamientos para aportar realismo al juego.

Hay que remarcar también que en ningún caso se pretende desarrollar una aplicación con fines lucrativos ni tampoco se pretende conseguir una gran difusión de la misma, por lo tanto los aspectos de distribución de la aplicación y monetización no se han contemplado a lo largo de este trabajo y no se han tenido en consideración para ninguna de las decisiones tomadas.

1.2 Objetivos del trabajo

Como se desprende del punto anterior, el principal objetivo del trabajo es la formación personal en el ámbito de la ingeniería de *software* y el conocimiento de una tecnología extremadamente extendida a nivel mundial, como es Android y los dispositivos móviles modernos o *smartphones*.

Paralelamente y para satisfacer el objetivo principal se pretende completar el desarrollo de un juego de estrategia en tiempo real para dispositivos Android, tanto para *smartphones* como para tabletas. El juego final que se espera obtener consistirá en una versión básica del juego con funcionalidades

limitadas pero que sea completamente operativo y extensible en hipotéticas versiones futuras que aporten más dinamismo, jugabilidad y funcionalidad. El motivo por el que se plantea como objetivo una versión inicial del juego radica principalmente en ajustarse a los requisitos del trabajo de final de grado, dentro del marco de estudios de la *Universitat Oberta de Catalunya* y al número de horas computables a la realización del proyecto. Además es importante tener en cuenta que un juego de las características del que se pretende desarrollar implica un gran número de variables intrínsecas a la creación de juegos que no guardan relación con el mundo de la informática ni con el desarrollo de aplicaciones móviles. Adicionalmente estos proyectos implican a equipos interdisciplinarios; diseñadores, programadores, marketing, entre otros, competencias profesionales de las que se carece para este proyecto.

Concretamente nuestro juego está enmarcado en el desarrollo de una invasión alienígena. El jugador asumirá el rol de alienígena y su principal objetivo consistirá en la total y completa exterminación de la raza humana mediante aplicación de recursos que podrá obtener a lo largo de la partida. La metodología de juego escogida es similar a la de otros juegos que podemos encontrar en el mercado, como es el caso de "Plague", en el cual el jugador no tiene control directo sobre las acciones realizadas por el alíen sino que únicamente podrá tomar decisiones sencillas en torno al movimiento así como aplicar una serie de mejoras sobre el organismo de los individuos extraterrestres con el fin de mejorarlos y hacer viable su colonización. En base a todo lo expuesto se pretende que el usuario, a través de sus decisiones, condicione la colonización, pero a su vez la estrategia escogida por él será determinante para fijar las acciones adoptadas por la humanidad. De manera que las distintas estrategias de juego pueden variar significativamente el avance del juego, haciendo cada partida única.

El contexto del juego, sus objetivos, reglas y el resto de cuestiones relacionadas con él se tratarán con más profundidad en puntos posteriores destinados a ese fin.

1.3 Enfoque y método seguido

En lo referente al enfoque de la aplicación se ha tenido en consideración que existe un gran número de aplicaciones similares a la que se está desarrollando en este momento, lo cual constituye una valiosa fuente de información que podemos aprovechar. Sin embargo, pese a la existencia de juegos similares, éstos no son de código abierto ni se dispone de partes del código de la aplicación dado que los creadores los crearon con finalidades lucrativas, de modo que ha sido necesaria la realización del proyecto desde cero empleando en los casos en que se ha considerado oportuno aplicaciones y herramientas de terceros.

Siendo fieles a la realidad, el proyecto no ha sido desarrollado totalmente desde 0, puesto que prácticamente la totalidad, exceptuando los artefactos de negocio, de la aplicación se ha basado en el *framework* de Android así como

en las recomendaciones dadas por Google sobre éste, por lo que no se ha empleado programación C de ningún tipo y se ha trabajado casi exclusivamente en la capa de superior del *framework*, capa e aplicaciones.

Centrándonos en la metodología, cabe destacar que en el presente proyecto se ha seguido una metodología de proyectos tradicional iterativa por lo que el proyecto se dividirá en las siguientes cinco fases:

1. Iniciación
2. Especificación
3. Implementación
4. Control y testing
5. Cierre

Puesto que es un proyecto iterativo, las fases, especialmente la de implementación y control, no se ha ejecutado completamente de forma secuencial y se ha ofrecido *feedback* entre ellas a lo largo de todo el proyecto para tratar de optimizar los resultados de cada fase.

Para las fases de iniciación y especificación, aprovechando que la presente aplicación tiene grandes similitudes con otros juegos ya existentes en el mercado se procedió a la realización de un análisis a fondo de algunos aspectos de la jugabilidad y de la interfaz de usuario (UI en adelante) de los mismos. Ello permitió extraer una serie de conclusiones que ha ayudado a fijar las bases y a establecer un punto de partida para el desarrollo del juego. Se consideró que el análisis de aplicaciones similares es una buena opción dado que permite observar el funcionamiento de alguna de las posibles soluciones a los retos del desarrollo que se nos planteaban, reduciendo el nivel de incertidumbre típico de las primeras fases de un proyecto, y evitar así caer en decisiones equivocadas que podrían haber sido detectadas de forma tardía causando un gran aumento en el coste de corrección.

Por lo que se refiere al desarrollo y a la arquitectura se ha seguido un planteamiento por componentes, el cual se definirá más detenidamente en la sección de este trabajo destinada a ese fin. Destacar que los componentes fueron planificados de forma global; definiéndose funcionalidad, casos de uso y alcance entre otros, y posteriormente se realizó un refinamiento, desarrollo, implementación y testing individualizados y secuenciales. No obstante siguiendo el patrón iterativo y en caso de resultar necesarias modificaciones en la programación, diseño o planificación éstas se han aplicado antes de proseguir con el siguiente componente. Este planteamiento ha permitido, por un lado, tener una imagen general del proyecto y las relaciones entre los componentes y al mismo tiempo la detección de posibles errores en el diseño general previa implementación.

Finalmente y teniendo en cuenta las limitaciones en tiempo, esfuerzo y conocimientos, se ha intentado seguir un patrón de diseño centrado en el usuario (DCU), aplicándose no solamente a los aspectos tecnológicos del proyecto sino también a la propia definición del juego y conseguir un producto final lo más ajustado a las necesidades de los usuarios finales potenciales.

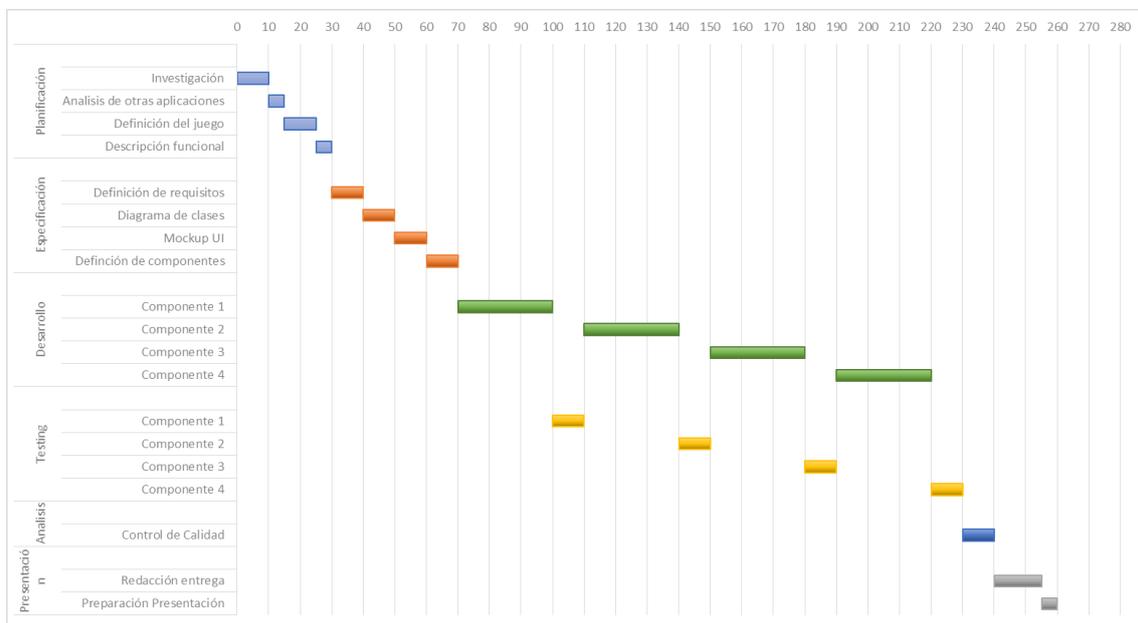
1.4 Planificación del Trabajo

La planificación de horas destinadas al desarrollo del TFG que se confeccionó fue la siguiente:

Días laborables: 2 horas diarias

Días festivos y fines de semana: 5 horas diarias.

Contando con trece semanas de trabajo, que se iniciaron en la primera semana de Octubre y sin contar los días del mes de enero que fueron reservados para imprevistos, se estimaron aproximadamente doscientas sesenta horas de trabajo, que se distribuyeron del siguiente modo:



Para realizar el seguimiento de la planificación se dividieron las fases del proyecto en diferentes tareas para así hacer más sencilla y precisa la estimación de su coste y a su vez facilitar el seguimiento de la planificación establecida. Con ese mismo fin se evitaron en todo momento posibles solapamientos entre las tareas ya que por un lado no tiene demasiado sentido al tratarse de un proyecto desarrollado por una misma persona y por otro que dificulta conocer el progreso real de cada una de las fases al no ser tan identificable visualmente sobre el diagrama de Gantt anterior. Diagrama que permitió identificar fácilmente el punto exacto en el que nos encontrábamos en todo momento durante el desarrollo del proyecto. No obstante, es importante remarcar que las estimaciones iniciales que se muestran en el Diagrama de Gantt resultaron ser incorrectas en algunos puntos, lo cual ha supuesto la necesidad de realizar algunas modificaciones sobre el mismo, tales como la eliminación de los test unitarios automáticos y como consecuencia de ello han tenido que ser limitados a test funcionales de integración y recursión.

Del mismo modo que se tuvo que limitar la profundidad de los test realizados también fue necesario emplear, en su totalidad, los días del mes de enero reservados como parte de la gestión del riesgo. Adicionalmente también se suprimió la fase de control de calidad y se sustituyó por una encuesta sobre los usuarios de la aplicación idéntica a la empleada en la fase de planificación, sin tener como objetivo la modificación del producto, puesto que la extensión del juego con futuras versiones no se encontraba dentro del marco del proyecto, pero sí han servido como valoración objetiva de la aplicación y han posibilitado la obtención de información para extraer las conclusiones finales.

1.5 Breve descripción de los productos obtenidos

El producto principal obtenido es un juego con funcionalidad básica y con la posibilidad de mejoras a través de versiones y extensiones. Al margen de este producto se ha obtenido los librables típicos de la gestión de proyectos que se ven recogidos de forma sintética en esta memoria. Adicionalmente se ha obtenido información sobre preferencias en materia de juegos en dispositivos móviles a través de las encuestas realizadas y algunos artefactos de diseño; imágenes principalmente.

1.6 Breve descripción del resto de capítulos de la memoria

La memoria se ha estructurado siguiendo un acercamiento de arriba hacia abajo, de forma que se iniciado con información general y a lo largo de las secciones se puede ver con más detalle algunos de los aspectos más relevantes. Se ha elegido esta organización por dos puntos: Poder comprender mejor conceptos que puede resultar confusos fuera de contexto y seguir el modelo planteado para el proyecto.

Los capítulos son:

2. El Juego: A lo largo de este capítulo se definirá el juego que se ha desarrollado, sin pretender entrar en aspectos tecnológicos o de diseño y con la mera intención de que se comprendan correctamente de las decisiones tomadas en posteriores capítulos.

3. Planificación: En esta sección se abordará toda la fase de obtención de información de los usuarios y de otras aplicaciones existentes, se justificarán las decisiones tomadas sobre el planteamiento inicial del juego y se definirá la funcionalidad del juego.

4. Diseño: En este capítulo se definirán los componentes, describirán los aspectos de diseño técnico y arquitectura.

5. Implementación: Se pondrá especial énfasis en la implementación de la lógica propia del juego, haciendo mención a elementos que hayan sido

relevantes por su repercusión sobre la aplicación global o por la complejidad y problemática encontrada.

6. Conclusiones: Se expondrán las conclusiones generales obtenidas del proyecto y se podrán en contraposición con los objetivos planteados al inicio del proyecto.

2. El Juego

2.1 Objeto del Juego

A.Invasion está englobado dentro de lo que se conoce como juegos de estrategia, en concreto se trata de un juego incremental o matemático. El juego se desarrolla en el marco de una invasión alienígena en la Tierra, asumiendo el jugador el rol de alien. El jugador por tanto, persigue el fin de exterminar a la raza humana, para lo cual deberá proceder a la colonización de la totalidad de los territorios existentes partiendo de un país de origen en el que han aterrizado los extraterrestres, país que será elegido por el propio jugador y dicha elección constituirá el momento inicial de la partida del cual surge el primer alien.

Se ha planteado el juego de modo que el usuario no tenga control directo sobre los individuos alienígenas ni sobre sus acciones, sino que el jugador simplemente deberá adoptar una serie de decisiones sencillas, que consistirán:

1. Decidir si se mueve de un territorio a otro.
2. Mejorar los organismos de los aliens para que la colonización resulte más viable.

Hay que destacar que todos los cálculos matemáticos que se exponen a continuación se llevan a cabo a nivel de región, es decir que se tienen en consideración el número de alien de una región y no el global, del mismo modo que sucede con los atributos de los humanos. Esto es así siguiendo modelos biológicos que suponen poblaciones aisladas ya que sino algunos aspectos como el movimiento tendrían poco sentido si se considerara la población alienígena como continua. No obstante, hay que destacar que los atributos de los alien sí son globales, esto es así para simplificar el juego ya que un control de los atributos por región supondría una complejidad técnica y de la dinámica del juego totalmente fuera de los objetivos del presente proyecto.

2.2 Skills

Para producir estas mejoras deberá incrementar de nivel de sus habilidades. En concreto va a poder incrementar cuatro tipos de habilidades:

Habilidades relativas a su reproducción: La mejora de éstas implicarán una mayor capacidad de generación y por tanto supondrá la producción de un mayor número de individuos alien.

Habilidades relativas a su movilidad: Su incremento conllevarán un aumento de sus capacidades de propagación por el territorio y por ende podrán alcanzar mayor distancia en un tiempo menor.

Habilidades relativas a su ataque: La mejora del ataque supondrá una mayor capacidad de matar a los humanos.

Habilidades relativas a su defensa: El aumento de los sistemas de defensa implicarán una mayor resistencia a los posibles ataques de los humanos, y por tanto convirtiendo sus organismos en más resistentes.

Es importante tener en cuenta que las mejoras a las habilidades van a tener un coste determinado, en concreto cada mejora que se realice va suponer el gasto de un determinado número de puntos de evolución. Estos puntos de evolución se obtendrán a lo largo de la partida como retribución al jugador por el avance de su colonización, tal y como expondremos más adelante.

Ya hemos señalado que las mejoras se van a poder realizar sobre cuatro habilidades básicas, pero es necesario saber que dentro de cada una de las tipologías de habilidades a mejorar encontraremos distintos grados, en concreto se ha considerado que cuatro niveles de mejora por cada habilidad resultaban suficientes. Es evidente que el número de puntos de evolución para subir de un rango a otro será cada vez mayor ya que los beneficios que aportarán a las skills de nuestro alien también aumentarán sustancialmente con cada nivel.

2.3 Puntos de evolución

Centrémonos ahora en los puntos de evolución. Como hemos indicado estos se obtendrán por el jugador como recompensa por el avance de la colonización alien en la Tierra, en concreto se obtendrá un punto de evolución cuando se dé alguna de estas tres circunstancias:

1. Duplicar el número de individuos alienígenas en la Tierra, lo cual permite obtener puntos de evolución rápidamente al inicio de la partida.
2. Invadir una nueva región.
3. Eliminar a la totalidad de los humanos de una región.

De manera que como se desprende del apartado anterior estos puntos actuarán a modo de moneda, para así poder establecer un criterio tangible y computable de la evolución y avance del juego por parte del jugador.

2.4 El mapa

Como hemos indicado anteriormente, la Tierra se encuentra dividida en regiones completamente diferenciadas, las cuales tienen características propias. Estos atributos están basados en datos reales, como son su superficie y población. En cada región los humanos dispondrán de una serie de atributos que son desconocidos para el jugador, y que también están basados en datos reales, como son:

Defensa: Cada región tendrá una defensa específica que se determinará en cada caso concreto de una manera fija basándose en criterios militares tales como su afiliación a la OTAN así como en criterios de carácter económico, de modo que la resistencia que encontrarán los aliens para la colonización será distinta en cada región.

Alerta: Ésta determinará la pasividad o agresividad de los humanos que en ella viven y se fijará en cada región atendiendo al número de humanos aniquilados que hayan dejado los aliens en cada caso; de modo que a mayor número de humanos muertos mayor será el nivel de alarma.

Ataque: Cada región tendrá un ataque específico que se determinará en cada caso concreto de una manera fija basándose tanto en criterios militares como en criterios de carácter económico, y que a su vez se encontrará influenciada por la visibilidad del alien en cada momento concreto.

2.5 Movimiento

Como hemos dicho que había regiones evidentemente será necesaria una forma de moverse entre ellas, para movernos entre los distintos territorios se ha utilizado una fórmula, basada en un modelo simplificado de probabilidad de migración en poblaciones naturales adaptado para ser jugable, principalmente por una reducción importante en el resultado final.

Es preciso destacar que la posibilidad de movimiento por el mapa no es libre y está condicionada al resultado de la siguiente fórmula:

$$\frac{(4 \times n \times pmov)}{100}$$

Dónde, n = número de aliens, pmov = probabilidad de movimiento (basada en atributo del alien).

Para tratar de plasmar el aspecto de probabilidad heredado de la fórmula original, el resultado obtenido se compara contra un número aleatorio, aumentado varios cientos de veces por motivos de jugabilidad, de modo la probabilidad de movimiento sea realmente una probabilidad. Adicionalmente se ha establecido otra limitación al movimiento que consiste en que cada región puede tener como máximo dos fronteras a las que moverse y además el movimiento a cada una de ellas solo es posible una vez. Con esta limitación se persigue limitar el número de puntos que puede ganar un jugador por este medio así como obligar a tomar una decisión estratégica antes de mover.

2.6 Crecimiento

El número de alíen se calcula en base a una formula simple para estimar el crecimiento de la población mundial, esta fórmula ha sido modificada para ajustarla a los objetivos del juego y permitir un crecimiento viable para nuestro propósito.

La fórmula utilizada es la siguiente:

$$n \times e^{trep \times t}$$

Dónde, n = población, e = constante (2,13..), trep = tasa de reproducción y t = tiempo (que en nuestro caso es 1 y por lo tanto la eliminamos).

Cada vez que se evalúa la formula se incrementará el número de alíen de forma automática. Como se puede apreciar en la fórmula, el modelo sigue un crecimiento exponencial, por lo que a mayor número de alíen mayor es el crecimiento por unidad de tiempo.

2.7 Muerte y condición de victoria

La muerte se calcula cada unidad de tiempo al igual que el resto de fórmulas, en este caso se realiza un cálculo para ambos bandos siguiendo las siguientes fórmulas.

La fórmula para calcular el número de humanos matados:

$$\frac{aatk \times n \times \rho}{hdef}$$

Dónde, aatk = ataque alíen, n = número de alíen, ρ = densidad de humanos y hdef = defensa de los humanos

La fórmula relativa al número de alíen muertros es:

$$\frac{\rho \times v \times hatk}{adef \times alarm}$$

Dónde, ρ = densidad de humanos, v = visibilidad, hatk = ataque de los humanos, alarm = nivel de alarma y adef = defensa de los alíen.

Podemos observar que se usa la densidad de los individuos en la realización de los cálculos con el fin de ajustarse a la realidad, puesto que se entiende que para poder matar es preciso que se encuentren. Para reflejar esa probabilidad de encuentro se ha usado la densidad de la población real, si bien es cierto que ésta no es precisa al 100%. La razón de usar este dato es que no tenía sentido entrar a valorar otros modos de cálculo que aunque resultaran un poco más precisos no aportarían un cambio sustancial en la jugabilidad y por tanto se ha considerado que el coste no merecía la pena.

Es preciso destacar que los alíen únicamente van a poder atacar en aquellos países en los que el nivel de alarma sea inferior a 5. Se ha considerado además que este nivel de alarma afecta al ataque ya que se presupone que los humanos en estas circunstancias están buscándolos activamente y que se encuentran con una preparación militar mucho más elevada.

Centrémonos ahora en la condición de victoria o derrota. Como se ha comentado anteriormente es preciso que los alíen maten a todos los humanos existentes en el mapa para poder alcanzar la victoria.

En este mismo sentido se ha establecido que la derrota se producirá en los supuestos en que los humanos maten a los alíen o en aquellos en que los alíen mueran.

Con la motivación de conseguir un juego que resulte más dinámico y divertido se ha introducido otra limitación, en este caso de tiempo, que puede conllevar la derrota. Es decir, se ha fijado un tiempo máximo para las partidas, en concreto de 700 segundos transcurridos los cuales, con independencia de la situación en la que se encuentre la invasión, dará lugar a la derrota, apareciendo un mensaje indicando que los humanos han sido capaces de desarrollar un arma biológica susceptible de matar a los alíen. Se ha adoptado esta decisión con el fin de incorporar presión a la partida y evitar cierto tipo de estrategias en el juego.

Por lo que respecta a la derrota simplemente añadir que en el juego no se ha incorporado como causa de derrota el hecho de que el jugador se quede sin movimiento para poder desplazar al alíen de una región a otra.

2.8 Extensión

Como hemos señalado en apartados anteriores a lo largo de este proyecto hemos desarrollado únicamente una versión básica del juego en la que no se han podido incorporar todas las variables. Sin embargo, el juego ha sido desarrollado con miras a futuras extensiones que permitan su incorporación y por tanto cuando se han planteado las fórmulas o las situaciones de las regiones ya se ha tenido en cuenta otras circunstancias, como es el caso del clima. El clima afectará a las condiciones del alíen para reproducirse y moverse en función de si está adaptado para soportar las condiciones meteorológicas y así se prevé la incorporación de un nuevo tipo de habilidades susceptibles de mejorar por parte del jugador, que son las habilidades de adaptación a cambios de temperatura.

También en futuras extensiones se prevé la incorporación de un nuevo dato que influya en los humanos para dar respuesta a los ataques alienígenas, que será la capacidad de comunicación del nivel de alarma y ataque entre distintas regiones.

3. Planificación

Siguiendo el modelo DCU en la fase de planificación se tendrá en consideración al usuario de modo que para la obtención de información se ha decidido emplear dos metodologías típicas diferentes; encuesta y benchmarking. Se ha considerado que la información obtenida con los métodos que implican al usuario puede resultar muy útil pero incompleta para el desarrollo de un juego, teniendo en cuenta que la base del mismo ya estaba previamente definida y además se encuentra limitada para encajar dentro del marco de un trabajo de final de grado, que implica serias limitaciones en los recursos técnicos, económicos y temporales disponibles para el desarrollo. Del mismo modo el benchmarking tampoco podía considerarse completo por sí solo.

Mediante estas dos metodologías podremos capturar las necesidades y preferencias de los usuarios, para poder desarrollar una aplicación que satisfaga el mayor número de necesidades del máximo número de personas. Al mismo tiempo nos permitirá también conocer nuestra audiencia potencial y desarrollar perfiles de usuario, que no solamente nos servirán para estructurar las necesidades sino que también aportarán valor a las pruebas de la aplicación final e identificarán necesidades típicas de ciertos grupos de usuarios.

3.1. Encuesta a usuarios

Otro de los métodos de análisis empleados fueron las encuestas realizadas a un determinado número de personas.

El método de la encuesta ha sido seleccionado principalmente por la simplicidad de distribución y la cantidad de datos que se pueden recopilar en relación al esfuerzo necesario. Quizás métodos como Shadowing podrían ser mucho más completos, pero resulta inviable asumir el coste temporal de hacer el seguimiento de varias personas durante un tiempo suficientemente prolongado y realizar la investigación con un solo individuo se obtendrían datos altamente subjetivos y difícilmente aplicables a un conjunto de usuarios.

3.1.1 Planteamiento:

La encuesta se ha elaborado con la herramienta Google forms (<http://www.google.com/forms>) ya que hace la creación del formulario bastante sencilla pero especialmente por la facilidad con la que se puede distribuir la encuesta a un gran número de personas, el seguimiento y recopilación de las respuestas y por la simplicidad con la que la gente puede acceder a la encuesta.

En cuanto al contenido, la encuesta se ha planteado con 15 preguntas más un campo de observaciones para permitir comentarios de los encuestados. Las

preguntas se plantearon con el objetivo de elaborar un perfil de usuario, especialmente haciendo énfasis en los contextos de uso.

Se tomó esta decisión basándonos en investigaciones previas de los hábitos de los jugadores en smartphones. Detectándose que los usuarios de móvil acostumbran a usar su dispositivo de forma combinada con otras actividades y en entornos muy variables, tratar de identificar estos contextos de uso se consideró la primera prioridad de la encuesta al determinar la cantidad de concentración y tiempo que los usuarios podrían invertir en el juego.

A continuación explicaremos brevemente el tipo de información que se pretendía conseguir con cada una de las preguntas:

Edad: Información demográfica, estrechamente relacionada con los conocimientos previos en nuevas tecnologías y hábitos de juego.

Preguntas 1, 2 y 3: Información sobre conocimientos previos tanto en juegos como en la plataforma móvil.

Preguntas 4, 5, 6: Información sobre contexto de uso.

Preguntas 7, 8, 9, 10, 15: Información sobre gustos del jugador, para identificar posibles mejoras de las mecánicas del juego

Preguntas 11, 12, 13, 14: Información para funcionalidades adicionales y diseño de Interfaces.

3.1.2 *Ánálisis de los resultados:*

Las respuestas se analizaron siguiendo el esquema anterior, hay que remarcar que los datos obtenidos son parcialmente sesgados debido a la distribución de la encuesta entre conocidos y amigos que mayoritariamente pertenecen al mismo rango de edad.

Datos demográficos: Las edades de los encuestadas se han englobado dentro de rangos de edad ya que se considera que el conocimiento de la tecnología está ligado a generaciones y no a edades concretas y de esa forma se consigue simplificar el análisis. Podemos observar que son utilizados fundamentalmente por gente joven, mayoritariamente menores de 30 años, y que las personas mayores de 50 no entran en el rango de personas que podrían estar potencialmente interesadas en nuestra aplicación.

Respecto a la necesidad que se busca cubrir con los juegos en Smartphones los resultados obtenidos son:

- a) Entretenimiento: 45%
- b) Que suponga un reto: 34%
- c) Competir con amigos: 22%

Este resultado es especialmente interesante, ya que como hemos planteado al inicio del trabajo, esa es fundamentalmente la necesidad que se pretende cubrir a través de *A. Invasión*.

Datos funcionalidades: En cuanto las funcionalidades se han obtenido los siguientes resultados:

1. Los jugadores no quieren aprender la mecánica de juego a través de manuales o tutoriales y quiere conocerla directamente jugando, un 50% de ellos con pistas y otro 50% por su cuenta.
2. El 62% de los encuestados encuentra los juegos por recomendación.
3. El 70% de los jugadores valora que el juego tenga una historia de fondo y de estos un 28% considera que es muy importante.

3.2. Benchmarking

Como acabamos de comentar el método de encuesta directa a los usuarios nos ofrece información incompleta para la información referente a la mecánica del juego ya que es complicado captar las necesidades y preferencias de un usuario que para comenzar no conoce el mismo. Con esa problemática en mente, se decidió crear una encuesta y unas pruebas guiadas con usuarios sobre aplicaciones ya existentes. Hay que destacar que la decisión de llevar a cabo el benchmarking con estas dos metodologías aportaba valor a otras fases al poderse emplear la misma en encuesta, concretamente se empleó la misma encuesta en la valoración de la interfaz de usuario, evidentemente sin las preguntas referentes a la jugabilidad, y en la valoración final de la aplicación. Con este planteamiento conseguimos tener unos valores de referencia sobre los que comparar las respuestas obtenidas para nuestra aplicación.

El motivo por el que se eligieron dos metodologías se debe principalmente a que no es posible realizar pruebas guiadas con un gran número de individuos, por varias razones entre las más importantes encontramos, la necesidad de ser presencial y la duración un poco más larga de la actividad. Estos dos factores son especialmente limitantes teniendo en cuenta los recursos de los que se disponía para realizar este trabajo.

Teniendo en cuenta que existen en el mercado aplicaciones similares a A. Invasión se consideró un buen método de análisis el facilitar el uso de dichas aplicaciones a distintos jugadores y posteriormente solicitarles que cumplimentaran una encuesta con diferentes preguntas relativas a la interfaz y al funcionamiento de los juegos con el fin de poder observar los puntos fuertes y débiles de cada uno de ellas.

A continuación detallaremos brevemente la información que pretendía extraerse con la encuesta:

Edad: Información demográfica, estrechamente relacionada con los conocimientos previos en nuevas tecnologías y hábitos de juego.

Preguntas 1 y 2: Información sobre conocimientos previos tanto en juegos como en la plataforma móvil.

Preguntas 3 a 8: Información sobre la interfaz.

Preguntas 9 a 16: Información sobre la funcionalidad del juego.

Preguntas 17: Información relativa a los gustos de los usuarios. Sin embargo, no debemos olvidar que esta información no tiene utilidad para el análisis de benchmarking dado que más de diez millones y un millón de descargas respectivamente en las aplicaciones analizadas son mucho más significativas que las conclusiones que puedan obtenerse de la presente encuesta.

Pregunta 18: Posible información adicional que puedan aportar los usuarios.

Lamentablemente no se ha extraído información relevante que mejore significativamente la que ya se obtuvo en la anterior encuesta. Pero si se ha obtenido información interesante en cuanto a posibilidades de juego y algunas ideas de diseño de UI que resultaron más atractivas para los usuarios de pruebas guiadas.

3.3. Escenarios de uso

3.3.1 Iniciar un juego nuevo.

Perfil que interviene: Jugador habitual, jugador casual

Contexto: El jugador se encuentra realizando otra actividad y tiene un descanso de algunos minutos que aprovecha para jugar una partida.

Objetivos: Entretenimiento.

Tareas:

1. Inicia la aplicación
2. Selecciona juego nuevo
3. Introduce un nombre para su alíen
4. Elige el tipo de juego (casual)
5. Juega
6. Cuando acaba su tiempo libre finaliza la partida

Información:

1. Necesita conocer información acerca de las características del modo de juego.
2. Necesita información durante el juego para conocer la mecánica.

Funcionalidades:

1. Poder iniciar un juego
2. Poder seleccionar el modo de juego
3. Poder jugar

Desarrollo:

1. Iniciar el juego: A través del menú principal
2. Elección del modo de juego y del nombre del alien: A través de las diferentes pantallas consecutivas

3. Juega: A través de la interfaz de juego y realiza las acciones necesarias mediante sus menús internos.

3.3.2 Continuar una partida

Perfil que interviene: Jugador habitual, jugador casual

Contexto: El jugador se encuentra realizando otra actividad y tiene un descanso de algunos minutos que aprovecha para continuar una partida que tenía a medias.

Objetivos: Entretenimiento.

Tareas:

1. Iniciar la aplicación.
2. Seleccionar continuar partida.
3. Jugar.
4. Finalizar partida

Información:

1. Necesita conocer información del juego durante la partida.

Funcionalidades:

1. Guardar datos partidas no finalizadas antes de cerrar aplicación.
2. Recuperar datos de partida no finalizada
3. Jugar.

Desarrollo:

1. Guardado de datos automático al finalizar aplicación sin haber acabado la partida.
2. Elección de continuar partida des del menú principal.
3. Acciones y funciones del juego des de los menús propios de la partida.

3.3.3 Consultar mejores puntuaciones

Perfil que interviene: Jugador habitual, jugador casual

Contexto: El jugador se encuentra realizando otra actividad y tiene un descanso de algunos minutos que aprovecha para continuar una partida que tenía a medias.

Objetivos: Reto personal, Competencia con amigos

Tareas:

1. Iniciar la aplicación.
2. Seleccionar mejores puntuaciones
3. Cerrar mejores puntuaciones.

Información:

1. conocer en base a que se ganan puntos.

Funcionalidades:

1. Calculo de puntuaciones.
2. Listado de las mejores 10 puntuaciones.
3. Almacenado de puntuaciones.

3.4. Perfiles de usuario

Ya hemos indicado anteriormente que los usuarios mayores de 50 años no son consumidores potenciales de nuestro producto, por lo que a continuación centraremos el análisis en los siguientes perfiles:

3.4.1. Jugadores casuales:

Edad: 20-50 años.

Destreza con Smartphones: Alta.

Conocimientos previos sobre juegos: Baja.

Contexto de uso: Son jugadores que habitualmente estarán realizando otras tareas y harán uso de la aplicación de forma simultánea o bien en sus ratos libres de modo que no reservarán parte de su tiempo libre para jugar, ni priorizarán el juego por encima de los sucesos que le rodean.

Tareas: Los usuarios únicamente deberán tener el juego instalada en u dispositivo, no se requerirá de conexión a internet ni de otras condiciones para poder jugar. Necesitarán ayuda para poder conocer el funcionamiento del juego ya que no son jugadores experimentados.

Elementos identificados: Para este perfil son la interfaz sencilla, la posibilidad de interrumpir el juego en cualquier momento y poder reanudarlo, sistema de consejos durante el juego.

3.4.2. Jugadores expertos:

Edad: < 30 años.

Destreza con Smartphones: Alta.

Conocimientos previos sobre juego: Alta.

Contexto de uso: Son jugadores habituales tanto en móviles como en otras plataformas, muy probablemente estarán realizando otras tareas de forma simultanea durante el juego o en momento libres durante las tareas, pero también jugaran durante tiempo de ocio sin grandes distracciones en su entorno, por lo que su tiempo de juego puede ser más prolongado así como su nivel de atención será mayor. Jugarán en varios lugares.

Tareas: Los usuarios únicamente deberán tener el juego instalada en u dispositivo, no se requerirá de conexión a internet ni de otras condiciones para poder jugar. Necesitarán alicientes para encontrar atractivo el juego a largo plazo.

Elementos identificados: Para este perfil son el sistema de puntuaciones, la compartición de puntuaciones, historia de fondo, modo de juego realista.

3.5. Conclusiones

Las conclusiones obtenidas a través de las encuestas son las siguientes:

3.5.1 Demografía y conocimientos previos:

Se determina que las personas de más de 50 años no son usuarios potenciales de la aplicación. Respecto a los otros rangos de edad de los que si se tiene información vemos que como norma general no son jugadores en otras plataformas pero sí podemos esperar amplios conocimientos del manejo de Smartphones. Por lo que la interfaz general de la aplicación debe ajustarse a los estándares de mercado, empleando iconografía estandarizada así como la distribución de los menús habitual, de ese modo nos aseguramos mantener la usabilidad para nuestros usuarios potenciales.

3.5.2 Contexto de uso:

Podemos observar que los usuarios utilizan el teléfono en una gran multitud de ambientes diferentes, por lo que se tiene que desarrollar una aplicación que sea usable tanto en reposo como en movimiento. Además las situaciones en las que los usuarios juegan son con mucha diferencia “ratos muertos” es decir no podemos pretender que estén totalmente focalizados en el juego sino que muy probablemente estarán llevando a cabo otras actividades de forma paralela o quizás perderán el interés en la aplicación en cualquier momento por agentes externos; anuncios en la tele, esperando a alguien, descargando programas, etc.

Finalmente el tiempo ininterrumpido de juego se encuentra dividido entre 5-10 minutos y más de 40.

Por todos esos motivos se toman las siguientes decisiones:

- El juego debe poder ser interrumpido y reanudado en cualquier momento.
- Se establecerán dos modos de juego; realista y casual. En el juego casual, la duración media de la partida deberá estar entorno los 10 minutos, mientras que en el modo realista las partidas tendrán una duración mayor añadiendo modificaciones a la mecánica básica del juego.
- Las interfaces de usuario así como la cantidad de información que debe manejar el jugador será lo más reducida, sintética y clara posible, de forma que el jugador pueda alejar su atención del juego parcialmente sin perder el hilo de la partida.

3.5.3 Preferencias respecto al tipo de juego:

Los encuestados prefieren juegos de mecánica sencilla y diferentes opciones de juego aunque también hay un número importante de respuesta que buscan juegos complejos, sin embargo por cuestión de tiempo y programa la posibilidad de aumentar la complejidad de las mecánicas de juego queda totalmente descartado. No obstante, se invertirán esfuerzos en ofrecer el máximo número de opciones de juego para tratar de satisfacer a los usuarios finales.

En cuanto a la necesidad de cubrir, un grupo considerable de encuestados buscan juego que supongan un reto, con esos perfiles en mente se construirá un sistema de puntuaciones y se planteará como futura extensión del juego la posibilidad de compartir las puntuaciones con amigos para tratar de abarcar los 3 perfiles de jugador encontrados.

3.5.4 Respecto a funcionalidades:

Se ha detectado que los encuestados desean conocer las mecánicas de juego durante el propio desarrollo de la partida, por ese motivo se construirá un sistema de consejos que irán guiando al jugador sobre las opciones que puede realizar. Estos "Tips" podrán ser ocultados durante la partida para tratar de satisfacer a aquellos usuarios que no los deseen o que ya conozcan el sistema de juego.

3.5. Casos de uso y flujos de interacción

Durante la fase de especificación se identificaron los siguientes casos de uso para la aplicación;

- CU_1. Un usuario quiere ver las mejores puntuaciones.
- CU_2. Un usuario quiere salir del juego.
- CU_3. Un usuario quiere jugar una partida.
- CU_4. Un usuarios quiere mover el alíen a una nueva región.
- CU_5. Un usuario quiere comprar nuevas skills.
- CU_6. Un usuario quiere finalizar el juego.
- CU_7. Un usuario quiere continuar una partida.

Dentro de estos 6 casos de uso encontramos tres de ello que están estrechamente ligados con la navegación entre las diferentes partes de la aplicación mientras que las otras tres están vinculadas dentro del juego y solo pueden suceder durante la partida.

Por motivos de síntesis los tres primeros casos y el caso CU_6 no se consideran oportunos de explicar ya que su flujo y el caso de uso es

extremadamente simple y solo consisten en navegar de una pantalla a otra y se pueden apreciar fácilmente con el siguiente diagrama:

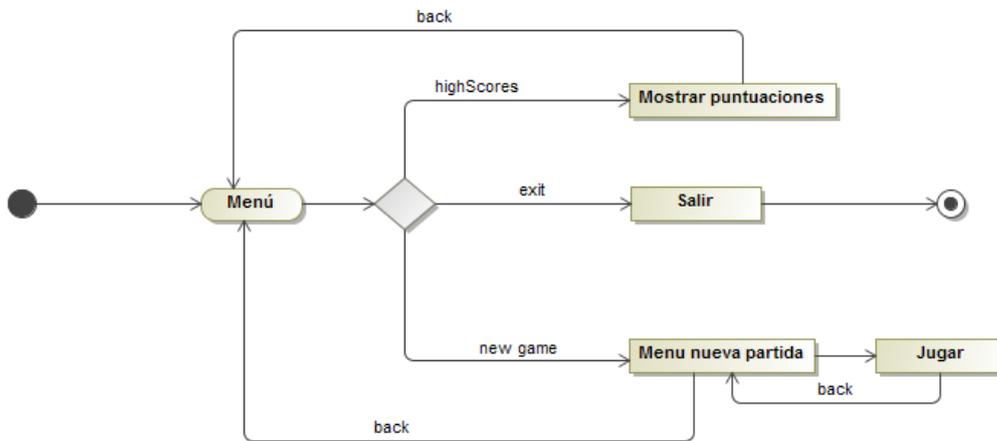


Ilustración 1: Flujo entre pantallas

3.3.1 CU_4: Mover alíen a una nueva región

Cuando un jugador tenga la opción podrá poner nuevos alíen en otra región del mapa:

Pre-condición: Para poder realizar un movimiento es necesario que la región con alíen tenga regiones colindantes sin alienas y disponga de movimientos.

Iniciado por: El cálculo de probabilidades de movimiento.

Flujo: Al usuario se le muestra un indicador de que tiene movimiento disponible y se espera su interacción. El usuario acepta el movimiento y este se procesa.

Post-condición: En la región elegida para el movimiento, que inicialmente no tenía alíen ahora sí los tendrá.

El flujo de este caso de uso es el siguiente:

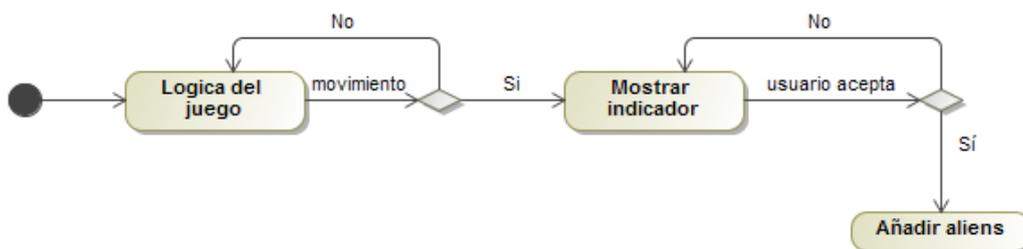


Ilustración 2: Flujo mover alíens

3.3.2 CU_5: Comprar nuevas Skills

El usuario decide comprar una skill nueva para mejorar el alíen.

Pre-condición: Debe disponerse de suficientes puntos de evolución para poder hacer la compra, la skill debe estar disponible.

Iniciado por: Usuario

Flujo: el usuario navega a la pantalla de compra de skills, selecciona una skill y la compra.

Post-condición: El alíen será el mismo que al inicio pero con sus atributos aumentados por la aplicación de la skill. El número de puntos de evolución disponibles será el mismo que al inicio menos el coste de la skill comprada. La skill comprada dejara de estar disponible y la siguiente skill del árbol pasará a disponible.

El flujo de este caso de uso es el siguiente:

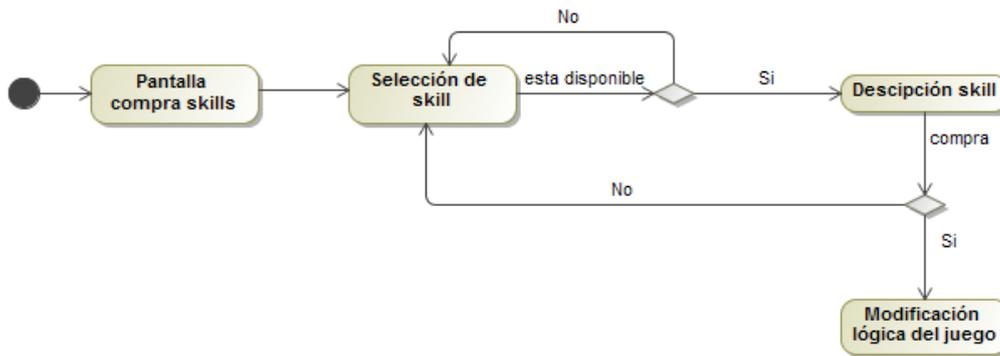


Ilustración 3: Flujo Comprar skill

4. Diseño

4.1 UI

Antes de empezar con el diseño técnico de la aplicación se realizó un diseño de la UI, ya que con los datos recogidos en la fase anterior de planificación y con los flujos de interacción teníamos toda la información necesaria para poder crear un prototipo de la aplicación, mediante el cual se obtuvo una guía para la implementación pero además un importante ayuda para el resto de la fase de diseño, al poder tener una imagen visual de los flujos de la aplicación.

Este desarrollo de la UI se llevó a cabo en dos fases, y mediante el uso de dos herramientas (ninjamock y Justinmind). Durante la primera fase en la que se utilizó ninjamock se creó un mockup completo de la aplicación por ser mucho más esquemático y sencillo, permitiendo la evaluación del mismo a través de usuarios sin necesidad de elegir colores e imágenes entre otros. De esa forma aplicar modificaciones aplicadas tras la evaluación fueron mucho más rápidas y además se ahorró tiempo en el diseño final. Para la segunda fase, se creó un prototipo final, con las imágenes, colores y diseño definidos totalmente, de forma que durante la fase de implementación no fuera necesario tomar ninguna decisión de diseño de UI y simplemente se llevara a cabo una “copia” del prototipo.

A continuación presentamos algunas de las decisiones más relevantes que se llevaron a cabo durante la definición y desarrollo de la interfaz de usuario:

4.1.1 Paleta de colores, imágenes y botones:

El diseño gráfico de la aplicación se realizó teniendo en mente que no es una competencia principal para un arquitecto de software y por lo tanto es transversal y menos importante y que las aptitudes personales son limitadas. Por ese motivo se optó en todo momento por hacer un diseño sencillo y básico considerándose que es preferible tener una interfaz clara y usable antes que intentar emplear gráficos complejos que además fueran deficientes en su resolución.

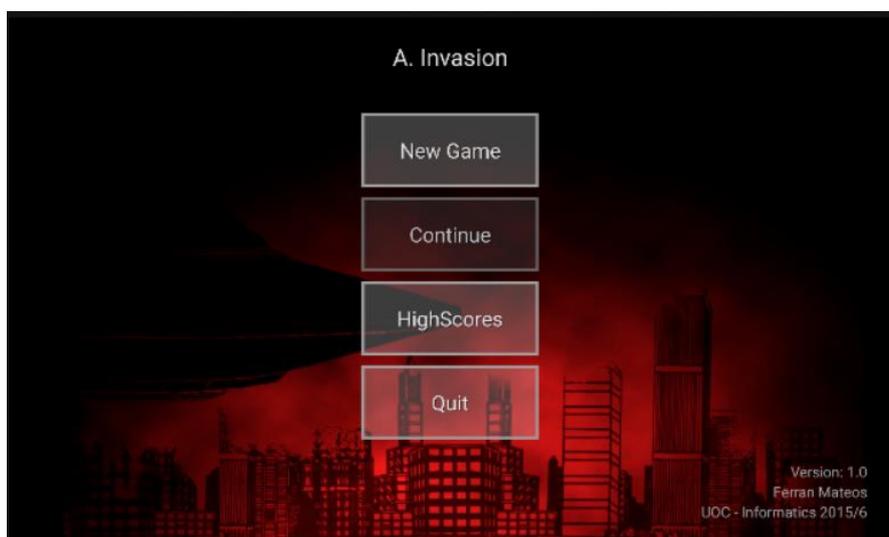
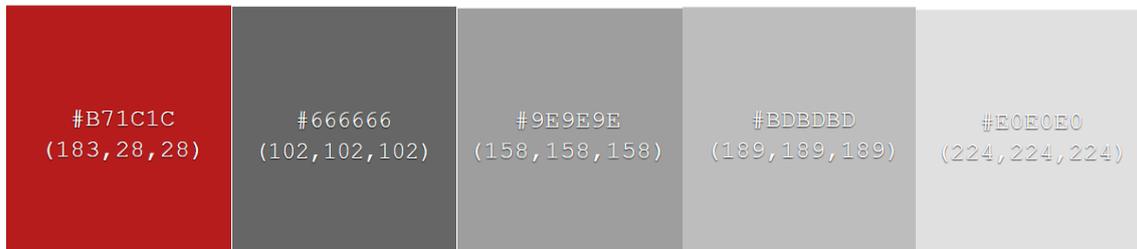


Ilustración 4: menú principal

Como se puede apreciar en la ilustración 4, se empleó una imagen de fondo representativa con la temática y un diseño de botones limpio y sencillo. La paleta de colores empleada a lo largo de la aplicación es la siguiente:



Usandose el color rojo como color para dar énfasis y característico de la aplicación y los cuatro tonos de gris como tonalidades para diferentes tipos de elementos: (de izquierda a derecha) deshabilitados, elementos de poca importancia, color secundario y color primario.

Se hizo esta elección de colores basandose en la guía de buenas prácticas de Google en la que recomiendan escoger tres colores de una paleta (similares) y un color de otra paleta como color para resaltar elementos.

Finalmente los botones son el elemento que más problemas presentaron a lo hora de decidir su diseño ya que, el diseño tradicional de los botones no acababa de verse atractivo con la imagen de fondo elegida y además se incurrió en una práctica poco recomendable a nivel de usabilidad; usar el mismo color para botones activos y para botones de navegación importantes. La decisión final fue optar por botones sin efecto al ser clicados ya que se dispone de un sonido que indica la pulsación y todos los botones mostraban vistas nuevas o pasaban a estado activo. Respecto a la mala práctica, durante la fase de pruebas de la interfaz no se detectaron grandes problemas de usabilidad por parte de ningún usuario por lo que no se estimó necesario hacer modificaciones.

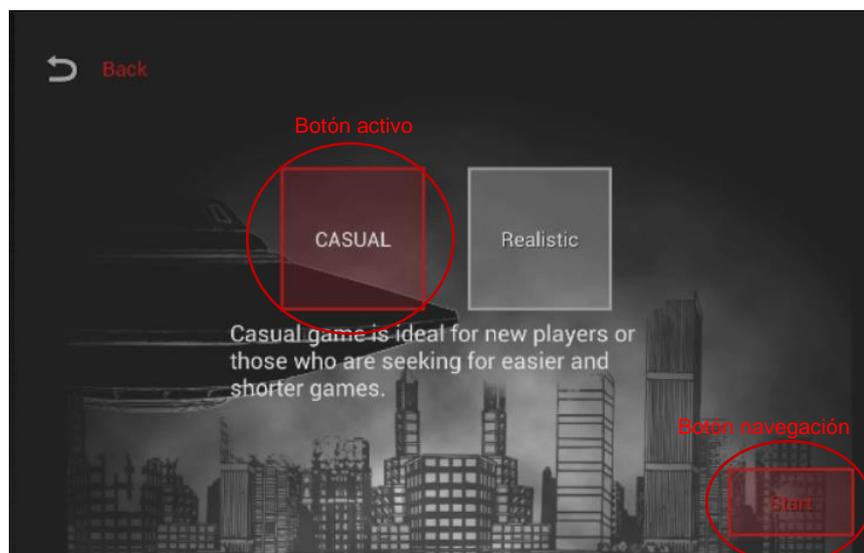


Ilustración 5: Color ambiguo de los botones

4.1.2 Imagen de juego:

Para la realización de la imagen del juego se encontraron problemas ajenos al diseño durante la implementación que obligaron a hacer modificaciones por lo que visualmente es menos atractivo del planteamiento hecho durante el prototipado, aún con eso, se mantuvo la distribución de elementos dentro de la interfaz; colocándose una barra inferior con la botonera para acceder a diferentes apartados y toda la información relevante contenida o dentro o encima de la misma de modo que el usuario no tuviera la necesidad de buscarla. Uno de los inconvenientes encontrados afecta a la posición del botón de volver que se quería colocar en la esquina superior derecha siguiendo el diseño que se había definido en el resto de pantallas, pero finalmente se decidió colocarlo en el la barra inferior, donde aunque no es un lugar habitual para colocarlo nos asegurábamos que el usuario lo vería o se dirigiría al estar allí el resto de controles:



Ilustración 6: Interfaz pantalla de juego

Finalmente comentar que a lo largo de la aplicación se decidió no emplear la barra de acción que proporciona el framework de android por dos motivos: primero en la imagen de juego nos ocuparía una sección importante de la pantalla y eliminarlo únicamente de esta pantalla pareció menos recomendable al romper el diseño general de la aplicación y segundo al no ser necesario un menú para ninguna pantalla, al tener una navegación secuencial, salvo en la pantalla de juego.

4.2 Componentes

Como se comentó en la introducción del trabajo, diferentes partes de la aplicación se dividieron en componentes. Estos componentes se crearon por agrupación de actividades similares pero al mismo tiempo también se crearon

con la intención de optimizar la planificación y seguimiento del proyecto. Con esa en consideración en mente se decidió dividir el conjunto del proyecto en 4 componentes:

Componente1: En este componente encontramos todos los elementos de navegación y sus funciones relacionadas, encontrándose dentro de este componente los casos de uso CU_1, CU_2 y CU_3; Ver las mejores puntuaciones, salir de la aplicación e iniciar una nueva partida. En este componente no se incluyó nada relacionado con el juego al requerir un arquitectura diferente y tener una complejidad más elevada.

Componente 2: Este componente encontramos los elementos relacionados con la persistencia, CU_7 y el guardado y recuperación de las puntuaciones.

Componente3: En este componente se encuentra la lógica completa del juego, salvando la función de guardado.

Componente4: En este componente encontramos las funcionalidades de navegación y representación de la información del juego. Estas funcionalidades se trataron de forma individualizada y no dentro del componente 1, debido a los motivos expuestos anteriormente.

4.3. Arquitectura y diseño

En esta fase del proyecto se definió toda la arquitectura siguiendo una arquitectura en capas concretamente el patrón MVC (Modelo – Vista – Controlador), donde el modelo contiene toda la lógica de negocio, la vista las representaciones visuales de la información y gestiona la interacción con el usuario y el controlador se encarga de hacer de nexo entre ambas partes. En nuestro caso prácticamente no tenemos capa de persistencia, que es la lógica que típicamente se encuentra en esta capa por lo que en el modelo no encontramos este tipo de implementación.

Se aplicó este patrón debido a que es el patrón que sigue el *framework* de Android. Sobre este último punto cabe decir que existe cierto grado de discrepancia, según las fuentes consultadas, sobre si el *framework* de Android debe ser entendido como MVC o como MVP, pero el debate no es de importancia para nuestro caso, especialmente teniendo en cuenta que MVP (Modelo Vista Presentador) deriva del patrón MVC y además considerando que el patrón se empleó para un primer diseño que posteriormente se refinó adaptándolo a la tecnología usada y por consiguiente modificándolo como resultó necesario.

4.3.1 Arquitectura componentes 1 y 2:

Para estos componentes se eliminó completamente la capa de modelo ya que con el uso de *Activity* proporcionado por Android y las *Views* estándares es posible ofrecer toda la funcionalidad de los componentes. En el caso del componente 2, la lógica de persistencia se encuentra acoplada con la lógica de

controlador, lo cual es una mala práctica dentro del modelo MVC, pero se consideró que crear un modelo completo para albergar la lógica de dos funcionalidades tan simples era un sobredimensionamiento del proyecto y por lo tanto se estimó conveniente dejar esa funcionalidad en la propia *Activity*, que actuaría a su vez como controlador y como modelo al ser empleada por otras actividades que necesitarán almacenar información.

4.3.2 Arquitectura componentes 3:

La arquitectura y diseño del componente 3 es sin duda la más compleja al albergar la capa de modelo con toda la lógica de negocio de la aplicación. En este componente, a diferencia de los otros sí fue completamente necesario definir un diagrama de Clases UML para poder planificar correctamente la implementación del mismo:

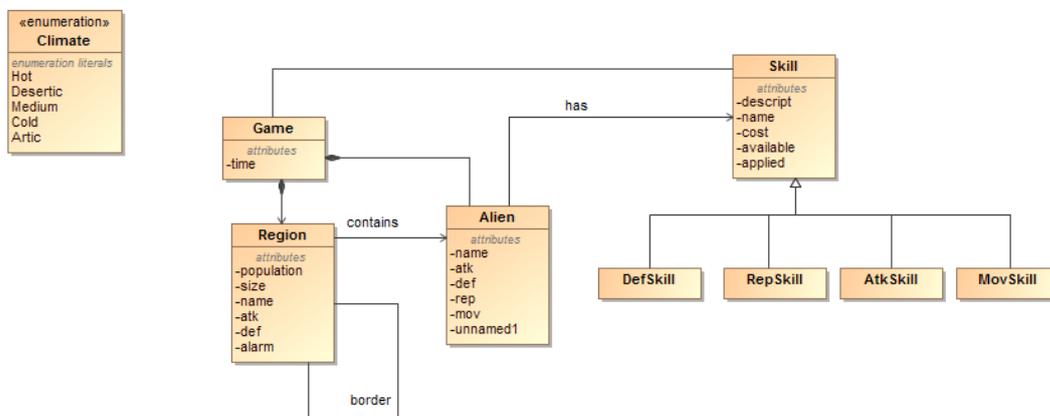


Ilustración 7: versión simplificada diagrama UML

Adicionalmente para este componente se se usó un patrón *Singleton*, en la clase Game, clase empleada como nexo de unión entre las diferentes clases y puerta de acceso a la lógica del juego. Mediante ese planteamiento y el uso del *Singleton*, el acceso a la información del juego se podía llevar a cabo en toda la aplicación.

La decisión de usar Game como “pegamento” de la lógica se tomó para mantener toda la lógica del juego completamente encapsulada e independiente del *framework* de Android ya que la implementación del componente 4 presentaba serias dudas sobre la mejor solución, del modo en que se planteado la lógica del juego se podría cambiar la vista y el controlador sin necesitare ninguna modificación sobre la lógica de juego, lo cual es lo que persigue MVC.

Finalmente respecto a la decisión de utilizar *Singleton* se adoptó esta medida en parte por la problemática comentada sobre el componente 4 y en parte por permitir la separación en diferentes actividades las vistas de skills y world, de modo que se evita enviar grandes mensajes entre las actividades con toda la información del juego.

4.3.3 Arquitectura componentes 4:

Este último componente es el que presentó más dificultades ya en la fase de especificación, debido al desconocimiento personal sobre el desarrollo en Android ya que no conocía el sistema previo a este trabajo. El problema principal al que hubo que enfrentarse con este componente fue la vista de un mapa interactivo el cual no se sabía cómo afrontar técnicamente y por lo tanto era muy complicado de refinar el diseño inicial, ya que a un nivel superior independiente de Android no se comportaba más que como otra vista con un controlador.

5. Implementación

En este capítulo no se pretende tratar temas de la implementación en su totalidad, primeramente porque no es el objetivo de la memoria de un trabajo final de carrera y en segundo lugar porque la implementación concreta puede ser observada en el código que se entrega conjuntamente con esta memoria. Por ese motivo en esta sección nos limitaremos a hacer mención a algunas partes de la implementación más relevantes y en especial a comentar alguno de los problemas que hemos encontrado.

Una decisión importante en esta fase fue la elección del SDK mínimo para el juego, en este caso se eligió hacer la implementación considerando como el mínimo la versión de Android API16, ya que las versiones anteriores a esta no tienen prácticamente presencia en el mercado actual según datos de Google. Aunque la versión mínima es API16, todo el juego está optimizado para dispositivos modernos y la mayor parte de las pruebas se han realizado con API 23 por lo que es la versión óptima para ejecutarlo. Del mismo modo que se da soporte a versiones anteriores también se ha hecho la implementación permitiendo una vista correcta en tabletas, aun así no ha sido nunca la prioridad. Respecto a la resolución de pantalla el juego en pantallas con una densidad de píxeles media o baja ya que las características del juego lo hacen inviable si el tamaño de las imágenes es demasiado reducido. En conclusión el juego está pensado para dispositivos de las últimas generaciones pero se ha intentado dar flexibilidad para que sea ejecutable en el mayor número de dispositivos posible.

5.1 SurfaceView

Tal y como se ha comentado brevemente en el apartado anterior la mayor dificultad encontrada con mucha diferencia ha sido la implementación de *SurfaceView*, se decidió utilizar *SurfaceView* implementando en el loop principal del juego, mediante la extensión de la interfaz *Runnable*. Con esta decisión se consiguió llegar a un producto funcional que ofrecía aproximadamente las funciones planificadas para el juego pero que rompía de una forma importante el modelo vista controlador, al quedarse *SurfaceView* muchas de las funciones que en principio deberían haber pertenecido a la *Activity*. Tratando de minimizar esta pérdida de funciones en el controlador se decidió que los menús se compondrían de nuevas actividades y situar todo el manejo de inicialización y fin de la vista de juego recayera sobre *Activity*. Una opción quizás más acertada para evitar esta pérdida del patrón MVC habría sido colocar el loop principal en la actividad.

La opción de implementar el loop en la *Activity* se descartó, primeramente por la cantidad de información que tendría que estar manejándose en una vista convencional de la actividad y por tanto ejecutándose en sobre el hilo de la UI, por otro lado las posibilidades de dibujar sobre un canvas que ofrece

SurfaceView parecieron las más apropiadas para poder implementar la funcionalidad necesaria.

6. Conclusiones

A lo largo de la realización de este trabajo se han aprendido cosas que personalmente considero muy importantes y que solo se adquieren a través de la práctica. Estas lecciones muestran un punto de vista que no se obtiene en ninguna otra asignatura a lo largo de todo el grado y son las siguientes:

- Conocimiento práctico sobre la gestión de proyecto, especialmente en aspectos relativos a planificación y gestión del riesgo: Aunque considero que mi planificación inicial tenía un grave error en el cálculo del coste, especialmente en lo referente al desarrollo del juego, he aprendido que una buena gestión del riesgo es importantísima ya que he tenido serios problemas con la planificación de todo el trabajo desde la segunda entrega al empezar a demorarse las tareas más de lo planificado y no haber hecho una estimación realista del tiempo que podía dedicar.
- Conocimiento sobre Android: Antes del inicio del proyecto no conocía la programación en Android ni tampoco la programación de juegos. Actualmente no me considero un experto pero creo que he alcanzado algunas competencias básicas bastante importantes para poder enfrentarme a futuros desarrollos en este entorno de forma mucho más cómoda y confiada. Valoro de forma especialmente positiva todo lo aprendido en este campo ya que era uno de mis objetivos principales para la elección de la temática del trabajo y considero que lo he cumplido bastante satisfactoriamente.

Respecto al alcance los objetivos iniciales, creo que he alcanzado positivamente todos los relacionados con los aspectos formativos, ya que como he comentado anteriormente entiendo que he aprendido los aspectos básicos del desarrollo de aplicaciones y de juegos así como creo que he aprendido muchas cosas referentes a la gestión de proyectos, aunque alguna de ellas las haya aprendido por lo que considero un fracaso en este proyecto, como es la planificación temporal.

En la línea de la planificación temporal creo que no he llevado una gestión correcta, sin embargo si considero haber gestionado el resto de aspectos del proyecto de una forma más o menos acertada como mínimo desde el punto de vista de ceñirme a la metodología definida, este hecho me ha hecho reconocer lo útil que puede llegar a ser algunas tareas que estudiadas de forma teórica su importancia puede resultar menos identificable. He de remarcar que la parte de control y testing no la he realizado como tenía previsto o me habría gustado no por alejarme de la metodología sino por una falta de tiempo derivada de los problemas antes comentados.

Finalmente y volviendo a los objetivos, estoy bastante satisfecho con la aplicación final obtenida, aunque sí me habría gustado poder añadir algunas funcionalidades que estaban previstas inicialmente y que por una cuestión de tiempo han tenido que ser recortadas, como el guardado de partidas, aplicar modificaciones según el clima, comunicación entre países y el lanzamiento de eventos automáticos durante el juego. Finalmente hay otras posibles extensiones sobre el juego final que no estaban contempladas y creo que

podrían haber mejorado mucho el juego, pero esas sí quedaban totalmente fuera del marco del proyecto.

4. Glosario

SO: Sistema operativo.

Estrategia en tiempo real: Término para referirse a juegos de estrategia donde el tiempo es continuo, en oposición a los juegos por turnos.

Framework: Anglicismo utilizado para referirse a un marco que define prácticas, conceptos, artefactos, módulos, etc. Que facilita una base y agiliza el desarrollo de programas.

Feedback: Anglicismo, retroalimentación. En el contexto de la gestión de proyectos hace referencia a que un fase ofrece información útil para otras anteriores.

Benchmarking: Anglicismo, hace referencia a proceso para evaluar dos productos similares.

Mockup: Anglicismo, maqueta o modelo a escala empleado para representar un producto real, habitualmente en demostraciones.

Prototipo: En informática, concretamente en diseño de interfaces se entiende como prototipo el diseño gráfico de una aplicación o programa que aunque puede contener alguna parte de programación relativa a animaciones, etc. No contiene ninguna lógica de negocio.

Capa: En el contexto del software se refiere a estructurar la lógica del programa en diferentes niveles donde cada nivel se encarga de una parte concreta.

MVP: similar al patrón MVC pero en el caso del presentador las vistas son pasivas y la interacción con el usuario recae en el presentador.

Singleton: en programación, patrón por el cual solo puede existir una instancia de una clase concreta en todo el programa.

5. Bibliografía

Webs

1. Información relativa a cálculos y formulas:

<http://www.coolmath.com/algebra/17-exponentials-logarithms/06-population-exponential-growth-01>

http://people.sc.fsu.edu/~pbeerli/pdfs/beerli_1998.pdf

http://www.researchgate.net/profile/Roy_Anderson2/publication/22682689_Population_biology_of_infectious_diseases_Part_I/links/0deec52bb8f565820d000000.pdf

2. Para la investigación previa relativa a la DCU:

<http://www.intexsoft.com/blog/item/179-statistics-of-the-mobile-game-development-market.html>

Consultada 3/10/2015

<http://cv.uoc.edu/webapps/xwiki/wiki/matm1202es/view/Main/5.2+Quan%3F>

Consultada 3/10/2015

<http://boxesandarrows.com/competitive-analysis-understanding-the-market-context/>

Consultada 15/10/2015

3. Información relativa a Android:

<http://developer.android.com>,

Consultada a lo largo de todo el proyecto.

<https://www.google.com/design/>,

Consultada a lo largo de todo el proyecto.

<http://www.hobo-web.co.uk/best-screen-size/>,

Consultada 30/10/2015

<http://gs.statcounter.com/#resolution-ww-monthly-201412-201505>

Consultada 7/11/2015

4. Stackoverflow, para FAQs y consultas de implementación:

www.stackoverflow.com,

Consultado durante todo el mes de diciembre.

Libros:

Android Game Programming by Example, John Horton. Packt Publishing Ltd. Junio 2015 Birmingham.

Beginning Android Games, Mario Zechner y Robert Green. Beginning Apress. Enero 2013

Otros recursos:

Para la realización de testing de fórmulas:

<https://jsfiddle.net/>

Códigos de color para paleta:

<http://www.color-hex.com/color/e0e0e0>