

Jesús Gallardo Santos
Grado de Ingeniería Informática
75.640 - TFG - Videojuegos

Ester Arroyo Garriguez
Javier Luis Cánovas Izquierdo

09/06/2019



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

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>TFG - SpaceBuddy</i>
Nombre del autor:	<i>Jesús Gallardo Santos</i>
Nombre del consultor/a:	<i>Ester Arroyo Garriguez</i>
Nombre del PRA:	<i>Javier Luis Cánovas Izquierdo</i>
Fecha de entrega:	06/2019
Titulación:	<i>Grado de Ingeniería Informática</i>
Área del Trabajo Final:	<i>75.640 - TFG - Videojuegos</i>
Idioma del trabajo:	ES
Palabras clave	<i>Puzzle, Android, Game</i>
Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i>	
<p>El objetivo principal de éste TFG es crear un videojuego cubriendo todas las etapas de su desarrollo y los roles profesionales implicados, hasta lograr un entregable completamente jugable.</p> <p>Para poder alcanzar las fechas propuestas para las entregas de la versión; parcial, jugable y final. Se ha optado por un juego de puzzle 2d para plataformas móviles, basado en una mecánica ya probada extraída de un juego de mesa.</p> <p>Usando un desarrollo basado en el modelo en cascada, centrandó la consecución de las principales tareas a partir de las fechas establecidas, se ha seguido un desarrollo a capas, donde aspectos poco relevantes como efectos gráficos, mejoras visuales, etc, se han relegado a fases finales del desarrollo.</p> <p>En general, el resultado final es de una calidad notable, teniendo en cuenta el tiempo invertido en el desarrollo y la variedad de roles que se han de cubrir (aspectos gráficos, sonido, programación, diseño, etc).</p> <p>Por tanto, SpaceBuddy es un juego de calidad que ofrece puzzles de alta complejidad y un tutorial para garantizar la comprensión de sus mecánicas principales, todo ello en un contexto simpático apto para todos los públicos.</p>	

Abstract (in English, 250 words or less):

The main objective of this FDP is create a videogame, and to cover all professional roles involved in the process, until achieve a fully playable game.

Focusing on the dates for the partial version, playable version and final version. A 2d Puzzle game for mobiles has been selected as best option, based on a previously tested board game mechanics.

Using a development based on the cascade model, the development has followed a layered develop, where most important points has been prioritized over other points relegated to the final steps, by example vfx, improvements, etc.

In general, the final result is of a remarkable quality, thinking about the small time available for the project and all involved roles covered (graphics, sounds, coding, design, etc).

Therefore, SpaceBuddy is a good quality game, that presents puzzles of high complexity with a smooth tutorial for learning main game mechanics, all over a enjoyable context based on space aliens, suitable for all audiences.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	1
1.3 Enfoque y método seguido.....	2
1.4 Planificación del Trabajo.....	3
1.5 Breve resumen de productos obtenidos.....	4
1.6 Breve descripción de los otros capítulos de la memoria.....	4
2. Estado del arte.....	5
2.1 Género puzzle 2d.....	5
2.2 La tecnología.....	6
2.2.1 GameMaker Studio 2.....	6
2.2.2 Construct 3.....	7
2.2.4 Unreal Engine 4.....	8
2.2.5 Unity3D.....	9
3. Definición del juego.....	10
3.1 Idea del juego.....	10
3.1.1 Breve descripción del juego.....	10
3.2.4 Objetivos planteados al jugador.....	17
3.2.5 Concept Art.....	17
3.3 Desarrollo y RoadMap.....	18
3.3.1 Evaluación de engines y kits de desarrollo.....	18
3.3.2 Planificación de objetivos.....	19
3.3.3 Cuantificación de tiempo y recursos por objetivo.....	19
4. Diseño técnico.....	23
4.2 Entorno de desarrollo.....	23
4.3 Software empleado.....	23
4.4 Assets.....	23
4.5 Arquitectura del juego.....	23
4.5 Importación del proyecto.....	27
5. Diseño de niveles.....	28
6. Evaluación del resultado.....	29
7. Manual de usuario.....	35
8. Conclusiones.....	35
9. Glosario.....	36
10. Bibliografía.....	37
11. Anexos.....	37

Lista de figuras

Ilustración 1: Planificación.....	4
Ilustración 2: Game Maker Studio 2.....	6
Ilustración 3: Drag&Drop GameMaker Studio 2.....	6
Ilustración 4: Construct3.....	7
Ilustración 5: Construct3 Web Editor.....	7
Ilustración 6: Unreal Engine 4.....	8
Ilustración 7: Blueprints.....	8
Ilustración 8: Unity3D.....	9
Ilustración 9: Unity3D Editor.....	9
Ilustración 10: Tablero Pelican Cove.....	12
Ilustración 11: Cartas de restricción orginales de Pelican Cove.....	12
Ilustración 12: SpaceBuddy - Menú de juego.....	13
Ilustración 13: Menú de configuración.....	14
Ilustración 14: Selección de nivel.....	14
Ilustración 15: Detalles de regla.....	15
Ilustración 16: Ayudas en pantalla.....	15
Ilustración 17: Nivel completado.....	16
Ilustración 18: Modo desafío.....	16
Ilustración 19: Concept Art.....	18
Ilustración 20: Versión Parcial.....	21
Ilustración 21: Versión Jugable y Entregable Final.....	22
Ilustración 22: Diagrama de flujo - modo aventura.....	24
Ilustración 23: Diagrama de flujo modo desafío.....	25
Ilustración 24: SpaceBuddy - UML entidades principales.....	27
Ilustración 25: Evaluador de mapas.....	28
Ilustración 26: Ejemplo configuración de mapa.....	29
Ilustración 27: Distribución de edad de usuarios encuestados.....	30
Ilustración 28: Nivel de estudios de usuarios encuestados.....	30
Ilustración 29: Feedback de usuarios - ¿Has completado el tutorial?.....	31

Ilustración 30: Feedback de usuarios - ¿Te han parecido claras las explicaciones de cada regla?.....	31
Ilustración 31: Feedback de usuarios - ¿Cómo evaluarías el nivel de dificultad de los mapas?.....	32
Ilustración 32: Feedback de usuarios - ¿Te parece suficiente el tiempo disponible?.....	32
Ilustración 33: Feedback de usuarios - ¿Te has sentido desorientado?.....	33
Ilustración 34: Feedback de usuarios - ¿te gustan los gráficos y animaciones?.....	33
Ilustración 35: Feedback de usuarios - ¿Te gusta la música del juego?.....	34
Ilustración 36: Feedback de usuarios - ¿Cómo de entretenido te parece el juego?.....	34

1. Introducción

1.1 Contexto y justificación del Trabajo

El desarrollo de un videojuego completo cubre todas las fases de desarrollo de software tradicional y engloba múltiples roles diferentes (Diseño, programación, arte, sonido, marketing, etc).

Por otro lado, en los últimos 50 años la industria de videojuegos ha evolucionado hasta convertirse en un sector con una recaudación mundial de más de 100 millones de dolares anuales.

Además, la barrera para producir y publicar juegos a nivel independiente prácticamente ha desaparecido. Steam en PC, Nintendo Switch en Consolas, Android Store y App Store en dispositivos móviles. Actualmente la mayoría de las plataformas son accesibles para el desarrollo independiente y cuentan con servicios que suponen una gran oportunidad, facilitando acceso inmediato a un enorme numero de consumidores potenciales.

Por tanto, elegir el desarrollo de un videojuego como TFG supone todo un desafío personal y una oportunidad de generar un producto actualmente muy demandado.

1.2 Objetivos del Trabajo

Tras la primera fase de análisis previa al desarrollo, en la que se analizó el objetivo principal del TFG, las posibles tecnologías y los MVP (*most valorated points*) de cada fecha, se definieron una serie de objetivos centrados en los siguientes aspectos:

- **Planificación:**

- Prototipado de todas las escenas y flujos del juego.

- División de aspectos esenciales y secundarios.

- Distribución de tareas en el tiempo respecto a las tres fechas principales.

- **Complejidad:**

- Definición del contexto de cada regla.

- Tutorial para presentar al jugador las mecánicas del juego de forma progresiva.

- Ayudas y controles para garantizar la comprensión de las reglas.

- **Contexto:**

- Diseño de elementos amigables que se adapten correctamente a las mecánicas principales del juego.

· **Desarrollo:**

Aprendizaje y dominio de las herramientas elegidas para el desarrollo del proyecto (Unity3D).

· **Adaptabilidad:**

División y orden de cada funcionalidad para adaptar el alcance añadiendo o quitando funcionalidad en función de la evolución del proyecto y tiempo disponible.

Evaluación constante de progreso y ajuste de planificación.

· **Verificación y Testeo:**

Diseño de herramientas y evaluación en usuarios sin conocimiento de la mecánica del juego, midiendo la percepción del nivel de complejidad y posibles áreas de mejora.

· **Documentación:**

Elaboración de documentos entregables en las diferentes fases de entrega continua.

Diseño y maquetación de videos de presentación.

Elaboración y evaluación de test de pruebas para usuarios.

Elaboración de memoria final del proyecto.

1.3 Enfoque y método seguido

El desarrollo de un videojuego por una única persona se convierte en un proyecto con un alto grado de dificultad debido al numeroso conjunto de perfiles involucrados.

Además, como parte del TFG vienen establecidos una serie de requisitos adicionales (entregas, documentación, videos de presentación, etc) que suponen un esfuerzo extra de planificación.

Por otro lado, uno de los aspectos más complicados en el desarrollo de un videojuego es el diseño de una mecánica nueva: debe resultar divertida, con grados de jugabilidad, autoexplicativa y finalmente, suficientemente probada, para garantizar que la mecánica está nivelada y resulte solida para poder desarrollar un videojuego sobre ella.

Por tanto, uno de los aspectos mas prioritarios al inicio del proyecto fue encontrar una mecánica ya probada, que pudiese portarse a un videojuego y sin competencia en el sector, reduciendo de éste modo los tiempos de diseño de mecánicas y los riesgos por falta de pruebas sobre esta.

De este modo, tras encontrar la mecánica en el juego de mesa Uluru, se obtuvo un conjunto de reglas en el que poder basar el videojuego mediante un nuevo contexto y algunas modificaciones para añadir más rejugabilidad, explicación y diversión.

Además, el tipo de mecánica elegida centra la complejidad del desarrollo en el código, uno de los aspectos más relevantes del perfil como alumno del grado en Ingeniería Informática.

1.4 Planificación del Trabajo

Durante la fase de planificación se definieron todos los aspectos fundamentales del juego; sus fases, estilos, escenas, flujos, niveles, etc. Así como la consecución de todos los objetivos respecto a las fechas fijadas. Todo ello se plasmó en el documento de diseño del juego GDD.

Además, como parte de la planificación se agrupó la funcionalidad en esencial y secundaria, permitiendo priorizar los aspectos más relevantes y listando la funcionalidad secundaria según su prioridad.

Por otro lado, aprender una nueva tecnología requiere tiempo y en lo concerniente a planificar sobre una tecnología desconocida aporta incertidumbre. Por tanto, durante el desarrollo se ha revisado constantemente la planificación para poder ajustar el alcance del proyecto añadiendo o eliminando aspectos en función de la evolución.

En la ilustración 1 de este documento se puede ver el diagrama de Gannt incluido en el GDD, donde se ve la división de prioridades y objetivos en función de las fechas clave.

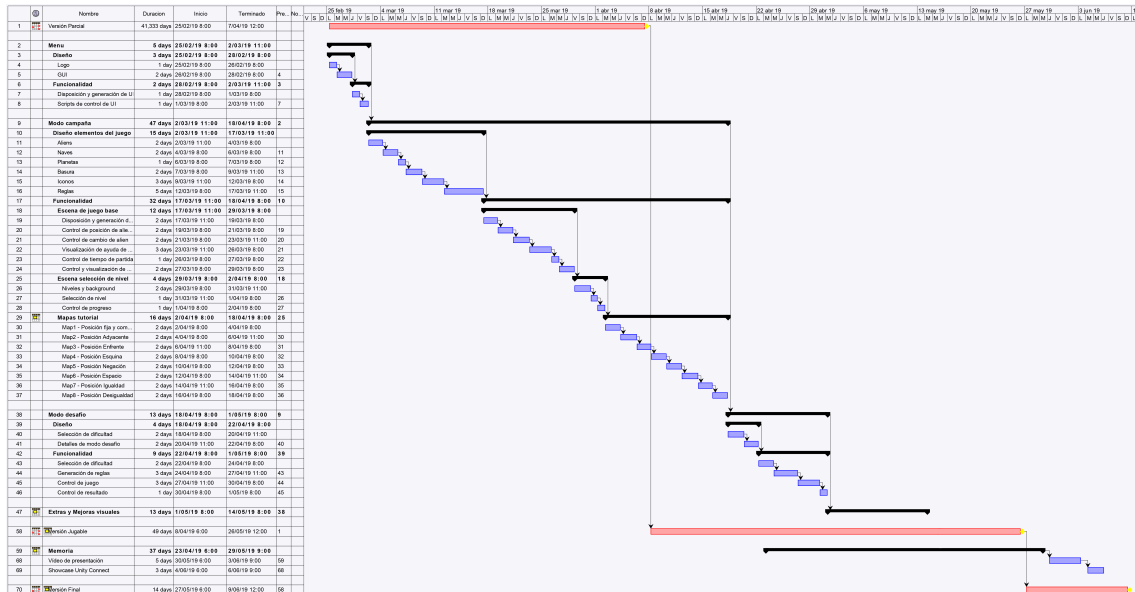


Ilustración 1: Planificación

En perspectiva, se sobrestimó los costes de la planificación, lo que ha permitido añadir funcionalidad adicional como la localización de textos, extras de gráficos y nuevos aspectos de jugabilidad.

La causa principal de este desvío en las estimaciones iniciales tiene dos factores: En primer lugar, se ha dedicado más tiempo de media al estimado y en segundo lugar, la incertidumbre de adaptarse al engine Unity3d y su editor que, finalmente, ha resultado una herramienta muy amigable con una curva de aprendizaje más suave de lo previsto.

1.5 Breve resumen de productos obtenidos

- APK versión *release candidate* para evaluación.
- GDD con el diseño y planificación del proyecto.
- LD con el diseño de niveles del proyecto.
- Memoria final del proyecto.
- Vídeo explicativo del concepto inicial del juego.
- Vídeo explicativo del proceso de desarrollo y aspectos fundamentales.
- Vídeo final de entrega – defensa y exposición del proyecto.

1.6 Breve descripción de los otros capítulos de la memoria

Durante los próximos capítulos se revisará el arte del juego: genero y tecnología utilizada. Además se entrará en detalle en el diseño técnico del proyecto y su

implementación, para finalmente comentar el producto generado, el feedback obtenido y dar paso a las conclusiones finales.

2. Estado del arte

2.1 Género puzzle 2d.

Los juegos de mesa son un entretenimiento ligado al ser humano desde los orígenes de la civilización. El primer registro de juego de tablero (Juego Real de UR) tiene más de 4500 años de antigüedad, hallazgo arqueológico de la civilización Sumiera.

De igual manera, a lo largo de los siglos se pueden encontrar diferentes juegos y evoluciones de juegos de mesa. Todos tienen en común que están destinados a entretener mediante el desafío de la mente planteando situaciones de lógica o puzzles.

Desde que en 1977 la firma Atari lanzase al mercado el primer sistema de videojuegos en cartucho, se pueden encontrar multitud de videojuegos que aplican este principio, situaciones en las que se plantean conflictos que se deben resolver mediante el ingenio.

Dentro de la evolución de los videojuegos podemos encontrar multitud de categorías. Además, conforme evoluciona la industria y se experimentan nuevos enfoques, las categorías se mezclan y surgen nuevas clasificaciones. No obstante, podemos decir que la categoría de puzzle contiene una serie de características comunes que la identifican:

- Problema inicial, plantea una situación de inconformidad que debe resolverse.
- Objetivo definido, se debe lograr un estado concreto considerado solución.
- Recompensa, obtenida mediante la consecución de los objetivos planteados.

Observando las tendencias del mercado de videojuegos y su evolución, podemos ver que se está buscando un público cada vez más casual y en general, se oferta más producto destinado al entretenimiento inmediato evitando las situaciones en las que el jugador obtenga desafíos mentales. Cada vez se aportan más elementos de acción y mecánicas que requieren un desafío de coordinación motora en lugar de ejercicios intelectuales.

Por tanto, se podría decir que el género de puzzle es poco demandado en general. No obstante, si se analiza el mercado de dispositivos móviles parece que la oferta de juegos de estrategia aumenta, quizás por el contexto en el que se utilizan estos dispositivos; partidas rápidas para entretenimiento inmediato e intermitente (transporte, esperas, etc), donde la interacción con el dispositivo ha de ser fluida y sencilla.

2.2 La tecnología

En la actualidad existen multitud de *engines* o *kits* destinados a simplificar la tarea del desarrollo de videojuegos.

En los últimos años se está viendo un aumento significativo de desarrollos *indies* debido principalmente a que los principales *engines* han incorporado licencias de uso de fácil acceso, pudiendo hacer un uso gratuito de las características del editor y realizando un pago proporcional al éxito del proyecto. Lo que supone en general, facilitar el acceso público a las herramientas.

Además, todos estos *engines* suelen generar con el mismo proyecto una solución para las diferentes plataformas: pc, consolas, móviles, etc. Por lo que se reducen mucho los costes de implementación por plataforma.

A continuación, una revisión general de las principales soluciones utilizadas en la actualidad para generar juegos multiplataforma y una breve descripción de sus prestaciones.

2.2.1 GameMaker Studio 2



Ilustración 2: Game Maker Studio 2

En lo concerniente a juegos 2D, GameMaker studio 2 es un motor pensado para reducir la programación, siendo posible incluso realizar proyectos enteros unicamente con drag&drop mediante la configuración acciones invocada por eventos. Dotando de todo lo necesario para que, sin conocimientos de programación, se puedan implementar un gran numero de situaciones que habitualmente serán suficientes.

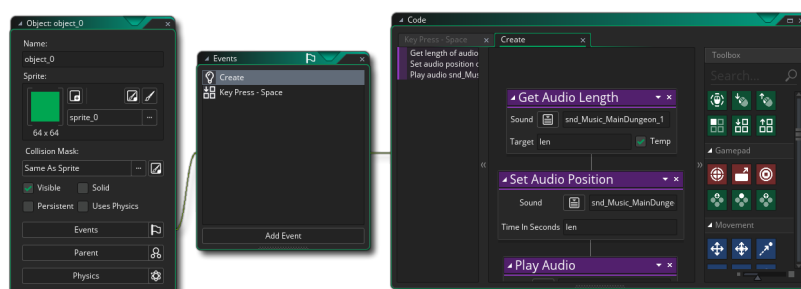


Ilustración 3: Drag&Drop GameMaker Studio 2

No obstante, además tiene su propio lenguaje de *scripting* llamado GML (Game Maker Language) que permite personalizar y extender las características base.

A pesar de ser multiplataforma pudiendo generar el juego para PC, consola y móviles, Game Maker realmente no compila el ejecutable. La implementación final en el dispositivo es mediante un interprete. Por ello, puede resultar menos eficiente en grandes proyectos respecto a otros competidores.

Por ultimo, su sistema anual de licencias por desarrollador tienen diferentes costes en función de la plataforma de destino, siendo 399\$ la licencia más cara destinada a juegos realizados para consolas (Xbox One, PS4 y Nintendo Switch)

2.2.2 Construct 3



Ilustración 4: Construct3

Este motor, orientado al desarrollo de juegos 2D, pone especial énfasis en el desarrollo para juegos de navegador con HTML5, donde el propio editor es una aplicación web.

Construct3 utiliza javascript como lenguaje base para el desarrollo, además, es extensible, permitiendo añadir plugins al editor.

Del mismo modo que GameMaker Studio 2, permite generar multiples soluciones con el mismo proyecto: PC, Consola y Dispositivos móviles. Pero de igual modo, se trata de un webview específico de la plataforma que carga el juego html5 generado.



Ilustración 5: Construct3 Web Editor

El sistema de licencias de Construct 3 es anual. Con un mínimo de 90€ para la versión personal, se desbloquean todas las características del editor.

2.2.4 Unreal Engine 4



Ilustración 6: Unreal Engine 4

Desde que en 2015 la compañía Epic Games liberase el uso del motor, cualquier persona puede utilizar libremente Unreal Engine 4 para desarrollar.

El motor y su editor permiten crear juegos de cualquier tipo de complejidad desde plataformas 2D a juegos VR y 3D triple A.

En el caso de Unreal Engine 4 es necesario conocer c++, aunque tiene un sistema de *blueprints* destinado a desarrollar de una forma más sencilla y visual, añadiendo funcionalidad a las clases existentes, pero este método no sustituye la programación.

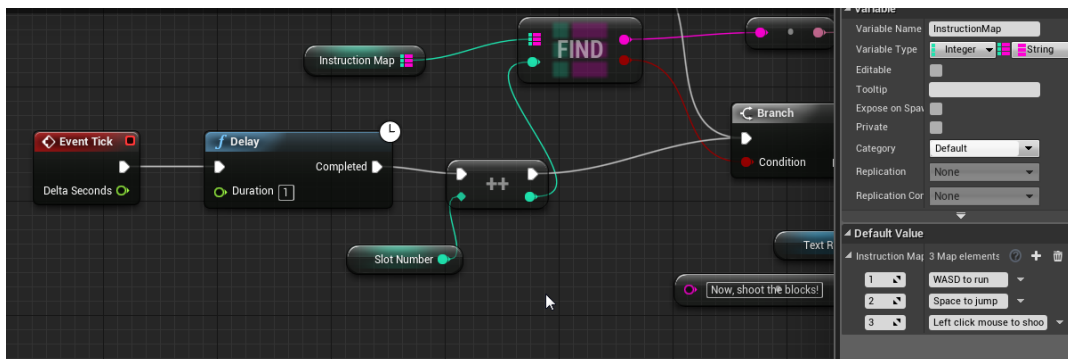


Ilustración 7: Blueprints

En el caso de UnrealEngine 4 las soluciones generados son compiladas de forma específica para la plataforma de destino y altamente eficientes.

A pesar de que su uso es gratuito, los juegos publicados con propósitos comerciales y realizados con unrealEngine, deberán pagar unos *royalties* del 5% de los beneficios trimestrales una vez sobrepasados los primeros tres mil dolares.

2.2.5 Unity3D



Ilustración 8: Unity3D

Probablemente el *engine* y editor del momento para el desarrollo Indie. Su principal ventaja es que es altamente versátil y potente tanto en proyectos 2D como 3D, pudiendo abarcar proyectos triple A de forma notoria.

Utilizando *c#* o javascript para el scripting, permite generar soluciones para múltiples plataformas de forma nativa. Además, dispone de un *Asset Store* y una comunidad muy grande que facilita el acceso a contenido.



Ilustración 9: Unity3D Editor

El editor de Unity3d permite drag&drop para la disposición de elementos y vínculos entre objetos, facilitando enormemente la labor pero sin ser un sustituto de la programación.

Unity3d tiene una licencia gratuita para uso personal que permite proyectos comerciales con beneficios inferiores a los cien mil euros anuales, si se excede dicha cantidad se debe abonar una licencia superior.

3. Definición del juego

3.1 Idea del juego

3.1.1 Breve descripción del juego

Se trata de la adaptación de una mecánica extraída de un juego de mesa llamado Uluru. En esencia, el jugador debe resolver puzzles aleatorios, donde se debe distribuir un conjunto de elementos, atendiendo a una serie de restricciones variables que condicionan las posibles posiciones de cada elemento. Convirtiendo su distribución en un desafío de lógica.

El juego tiene una ambientación simpática consistente en una vecindad espacial compuesta por 8 Aliens que tienen que distribuirse entre 8 naves/planetas. Para poder asignarle a cada Alien su correspondiente nave, el jugador deberá respetar las manías o deseos de cada uno, por lo que no todas las naves serán una elección correcta.

3.1.2 Subgénero y referencias a videojuegos existentes

SpaceBuddy es un puzzle en tiempo real donde lo importante es una mecánica, en concepto simple, que puede generar escenarios altamente complejos. Además, se han incluido mecánicas adicionales para aumentar la complejidad cuando se considera necesario.

No se han encontrado videojuegos con una mecánica similar a la que SpaceBuddy hace alusión. No obstante, es un genero donde existen multitud de alternativas que podrían presentar similitudes, en general: cualquier puzzle en el que se tengan que tomar decisiones de distribución atendiendo a reglas que dificulten la tarea.

3.1.3 Tipo de interacción juego-jugador

La interacción del jugador con el juego resulta simple, básicamente unos pocos *clicks(taps)* sobre los elementos para elegir que Alien posicionar y donde colocarlo, toda la dificultad recae la resolución del *puzzle*.

Durante la fase de juego, el jugador podrá:

- **Posicionar un Alien** mediante un *tap* en el Alien a elegir y otro *tap* en la nave destino.
- **Cambio/Cancelación de posición** al hacer *tap* en una nave destino, el Alien seleccionado dejará de estar disponible, si la nueva posición se encuentra ocupada, el actual Alien será devuelto al grupo de Aliens disponibles.

Se ha analizado la opción de arrastrar y soltar para posicionar los Aliens sobre las naves, pero en lugar de aportar más fluidez a la mecánica de juego, entorpece las interacciones en pantalla. Mediante *taps*, incluso los dispositivos con menor precisión resultan en una experiencia de juego aceptable.

Dado que se trata de una mecánica donde el desafío recae en comprender y satisfacer las restricciones impuestas a cada pieza y posición, el juego introduce cada regla de forma aislada, dedicando un espacio a su explicación y generando un entorno que garantice su comprensión. Por ello, se han diseñado niveles concretos en formato tutorial para cada regla. Este conjunto de mapas tienen habilitadas ayudas extra para facilitar la tarea en caso de que el usuario se sienta perdido.

Tras completar el modo aventura se podrá jugar el modo desafío, que genera niveles de forma procedural incrementando la dificultad a cada nivel superado. Por tanto, el jugador podrá poner a prueba su nivel de habilidad intentando alcanzar el máximo nivel posible de dificultad en el modo desafío.

3.1.4 Plataforma de destino

El tiempo disponible para cada puzzle, independientemente de su dificultad, es de 60 segundos. Teniendo en cuenta que la mecánica es simple así como la interacción en pantalla, supone un contexto ideal para dispositivos táctiles. Por tanto, la mejor opción es dirigirse a los dispositivos móviles, donde poder jugar unos pocos puzzles en minutos.

3.2 Conceptualización

3.2.1 Historia, ambientación y/o trama

Para la ambientación del juego se han elegido Aliens, y naves/planetas. El jugador deberá distribuir un conjunto de 8 Aliens en una agrupación fija de planetas atendiendo a una serie de restricciones que indican los deseos de cada Alien. Similar a establecer el orden de una colonia o la vecindad de los Aliens y sus posibles planetas.

Ejemplo: *“el Alien rojo quiere estar en uno de los tres planetas superiores, el Alien verde quiere estar adyacente al rojo.”*

De éste modo, el contexto ayuda a explicar todas las posibles restricciones y otorga una ambientación cerrada que permite acotar los recursos gráficos.

3.2.2 Definición de los personajes/elementos

Existen tres partes clave para la mecánica de este juego: elementos, posiciones y reglas.

- **Los 8 elementos a distribuir**

En la implementación están representados por 8 Aliens. Los Aliens son de diferentes colores para poder ser diferenciados y así identificarlos cuando unos tienen dependencia de otros.

- **Las posiciones fijas que pueden ocupar los elementos anteriores.**

8 naves y sus correspondientes planetas son las posiciones fijas que deberán ocupar los Aliens antes de finalizar el nivel.



Ilustración 10: Tablero Pelican Cove

- **Las reglas de disposición**

En la implementación, se identificarán los tipos de restricciones diferentes mediante 12 iconos únicos.



Ilustración 11: Cartas de restricción originales de Pelican Cove

Empezando por la esquina superior izquierda, de izquierda a derecha:

Carta 1	Comodín	El Alien puede ocupar cualquier casilla
Carta 2 a 5	Posición concreta	El Alien solo puede ocupar unas posiciones concretas.
Carta 6	Adyacente	El Alien quiere estar adyacente a otro Alien.
Carta 7	Enfrente	El Alien quiere estar enfrente a otro Alien.
Carta 8	Esquina	El Alien quiere estar esquinado con otro Alien.
Carta 9	Negación	El Alien no quiere estar ni en frente ni adyacente con respecto a otro Alien.
Carta 10	Espacio	El Alien quiere estar a dos posiciones de otro Alien.
Carta 11	Igualdad	El Alien quiere lo mismo que otro Alien.
Carta 12	Desigualdad	El Alien quiere lo contrario que otro Alien.

Se puede apreciar una complejidad incremental en cada tipo de regla y mediante la combinación del conjunto, se pueden obtener grandes desafíos.

Además, el tipo de mecánica identificada permite la ampliación de reglas y el control de su dificultad, así como la posibilidad de crear diferentes mapas predeterminados, estableciendo un flujo. Capacidades perfectas para la planificación y la expansión futura del juego si fuese necesario.

3.2.3 Interacción entre los actores del juego

A continuación se va a detallar el flujo de escenas en juego y sus interacciones.



Ilustración 12: SpaceBuddy - Menú de juego

El menú de juego solo permite acceder al modo desafío tras completar el modo aventura, destinado a comprender todas las mecánicas del juego. También permite acceder al menú de configuración.



Ilustración 13: Menú de configuración

El menú de configuración permite borrar todo el progreso, activar/desactivar el audio y modificar el idioma actual del juego.

Además, la ventana de configuración esconde un acceso al modo desarrollador. Para desbloquear el acceso al modo desarrollador es necesario pulsar repetidas veces el texto del título de la ventana “Configuración”.

El modo desarrollador permite evaluar las soluciones de cada mapa del modo aventura y desbloquear todos los mapas del modo aventura para dar acceso al modo desafío.



Ilustración 14: Selección de nivel

El modo aventura, está compuesto por 20 niveles. Cada nivel alterno se añade una nueva mecánica o regla de posición. Actualmente el modo aventura tienen una duración resumida para facilitar la evaluación y poder reducir los costes de desarrollo.

La versión comercial debería implementar más niveles de transición entre reglas, presentando una curva de dificultad más gradual y extendiendo la duración del modo aventura. En cualquier caso, presenta todas las mecánicas y niveles de transición suficientes para generar una experiencia de juego completa.

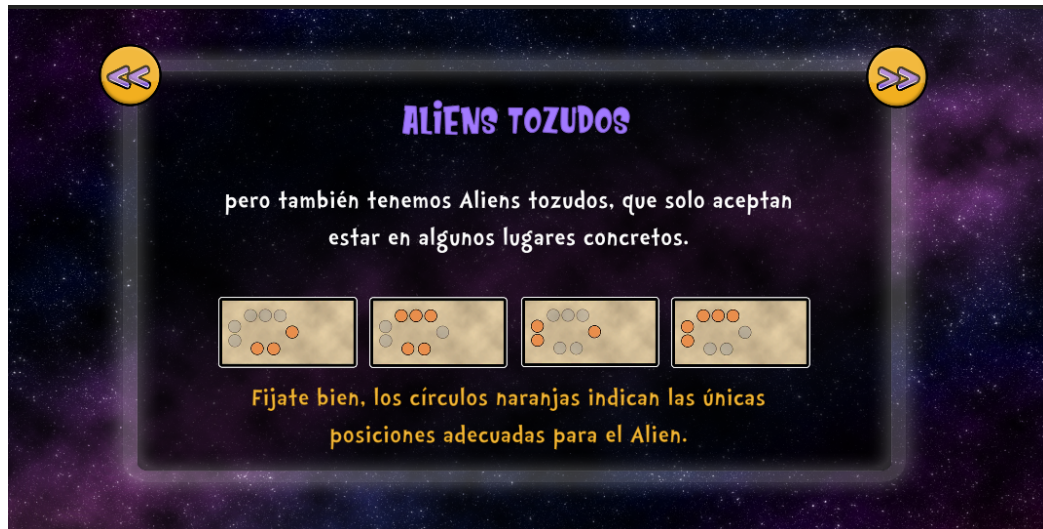


Ilustración 15: Detalles de regla

Cada vez que el juego introduce una nueva regla de posición, se dota de un contexto y explica la regla de forma independiente.



Ilustración 16: Ayudas en pantalla

Además, como parte de la comprensión de una regla nueva; durante el modo aventura, se muestran en pantalla las posiciones válidas de cada nueva regla, así como una guía para los primeros movimientos, reforzando la explicación inicial.



Ilustración 17: Nivel completado

Tras completar el nivel de forma satisfactoria (máximo un error), se desbloquea el siguiente nivel.



Ilustración 18: Modo desafío

El modo desafío consiste en una sucesión de mapas de dificultad incremental con mapas generados de forma procedural. Al inicio de cada nivel se muestra el número de posibles soluciones del nivel y la puntuación actual.

Durante el modo desafío, solo se permite continuar si no se comete ningún error. El jugador podrá retomar en cualquier momento el desafío anterior donde lo dejó o iniciar uno nuevo empezando de cero.

3.2.4 Objetivos planteados al jugador

El juego tiene dos modos:

- **Modo aventura**, compuesto por diferentes niveles de menor a mayor dificultad, destinado a asentar y garantizar la comprensión de la mecánica de juego y todas las restricciones.
- **Modo desafío**, genera mapas aleatorios de una dificultad incremental con el paso de los niveles. Permitiendo desafíos dinámicos en los que se tendrá en cuenta la mejor marca de tiempo obtenida.

Aunque la mecánica de juego resulta simple, aprender el funcionamiento de cada restricción y relacionarlo con su correspondiente icono, es el mayor desafío que tiene que resolver la implementación del juego. Por ello, el modo aventura pone especial énfasis en asentar las mecánicas implementando ayudas en tiempo de juego que poco a poco reduce.

Para ello, el modo desafío está bloqueado y solo se desbloquea tras completar la aventura, asumiendo que el jugador ya conoce y ha asimilado todas las reglas del juego.

3.2.5 Concept Art

En las figuras anteriores se muestra un pequeño prototipo de la distribución de los elementos en pantalla y estilo de arte. La estética elegida es un estilo *cartoon handpainting* con influencia de Francisco Ibáñez (conocido autor de los cómics: Mortadelo y Filemon)

Los principales objetos de juego (Aliens, naves y planetas) tienen animaciones que introducen movimiento para evitar el efecto pegatina que pueden producir las imágenes estáticas; Principalmente, gestos o micro movimientos para generar la sensación de vida en reposo de los elementos principales.

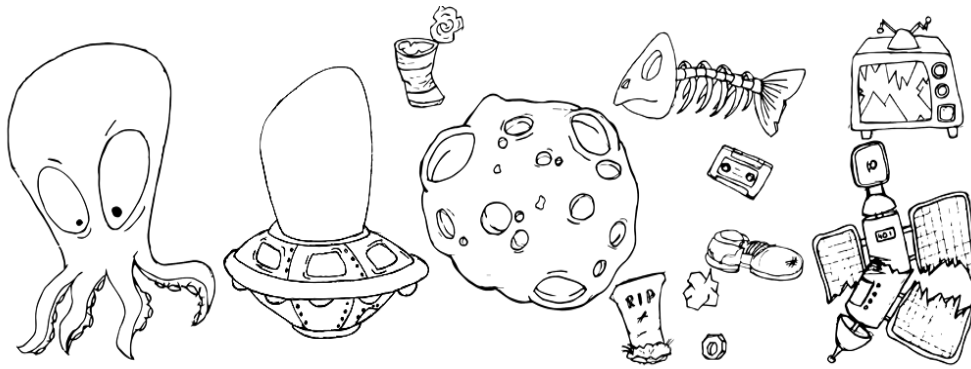


Ilustración 19: Concept Art

Además, sin llegar a sobrecargar la escena, cada planeta en uso por un Alien, generará basura espacial aleatoria, (por ejemplo: raspas de pescado, televisores, satélites, etc) para introducir un componente de humor y contribuir a la estética anteriormente mencionada.

En el análisis de desarrollo se ha contemplado cada uno de los recursos gráficos y sonidos que harán falta para cada escena del juego y se ha priorizado su diseño en función de su importancia, de forma que algunos componentes de mejora visual como los últimos mencionados, pudieran ser descartados de la versión final si se produjesen desvíos pronunciados en la planificación.

3.3 Desarrollo y RoadMap

3.3.1 Evaluación de engines y kits de desarrollo

Como se ha visto anteriormente, en la actualidad existen un gran numero de posibles soluciones para implementar juegos 2d cross-platform. Por tanto, para la elección del *Engine* conviene tener en cuenta otros aspectos personales como: el tiempo disponible para el proyecto, los conocimientos previos anteriores de los desarrolladores, preferencias, etc.

En lo personal, he dedicado algo de tiempo a Unity3D y me parece el *Engine* más amigable y con mejor curva de aprendizaje. Además, es uno de los *Engines* más utilizados en la actualidad para desarrollos *Indie* lo que permite encontrar gran cantidad de información que ayude a salvar los posibles obstáculos encontrados durante el desarrollo.

Por este motivo, el desarrollo se ha realizando con Unity3D para así poder minimizar el tiempo de aprendizaje y centrar los conflictos unicamente en el qué y no en el cómo.

3.3.2 Planificación de objetivos

La planificación se ha organizado en función de los 3 principales hitos o entregables.

- Versión parcial (entrega **07/04/19**)
- Versión jugable (entrega **26/05/19**)
- Versión final (entrega **09/06/19**)

Para la distribución de tareas, se ha tenido en cuenta una dedicación diaria de 3h de media en horario L-V 06:00h – 09:00h y S-D 10:00h a 13:00h. Además, se ha tenido en cuenta la carga adicional derivada de cada entrega (Empaquetados, Vídeos presentación, Memoria, *Showcase*, etc).

A continuación se detallan los principales objetivos, sus fechas y un breve comentario de las tareas acometidas.

“Para más detalle, el punto 3 de este documento incluye un diagrama de Gantt”

Menú y recursos base	Inicio: 25/02/19	Fin: 02/03/19	Dedicación: 5 días
Modo aventura	Inicio: 02/03/19	Fin: 18/04/19	Dedicación: 47 días
Modo desafío	Inicio: 18/04/19	Fin: 01/05/19	Dedicación: 13 días
Extras y Mejoras	Inicio: 01/05/19	Fin: 14/05/19	Dedicación: 13 días
Memoria	Inicio: 23/04/19	Fin: 29/05/19	Dedicación: 37 días

Durante la fase final previa a la entrega y en paralelo a la elaboración de la Memoria, se estimó una etapa de corrección de errores y mejoras de menor impacto en función del feedback recibido por parte de los usuarios de prueba.

3.3.3 Cuantificación de tiempo y recursos por objetivo

Finalmente, como parte de la planificación anterior, se realizó un diagrama de Gantt y se fijaron los MVP para cada entrega en Google Calendar. Ambas herramientas, básicas para medir el progreso del desarrollo y detectar cualquier desviación durante el proceso. Dado que no existía ningún otro desarrollador en el proyecto, se descartó usar un *canvas agile* usando Google Keep en su lugar para el control del TODO.

Además, se diseñaron medidas de actuación en caso de retraso en las estimaciones, identificando aquellos puntos que podían ser sacrificados sin comprometer la viabilidad del proyecto, tales como estética y decoración o mejorados en caso de contar con más margen, como traducciones o efectos extras.

Las previsiones iniciales contemplaban además un margen de seguridad para posibles contingencias no planificadas.

La dedicación final estimada al desarrollo del juego fue de 78 días (234h), con un margen de seguridad de 9,4% (12 días), antes de la fecha de entrega, Versión Jugable.

A continuación, se adjuntan los diagramas de Gantt de cada uno de los principales objetivos identificados anteriormente.

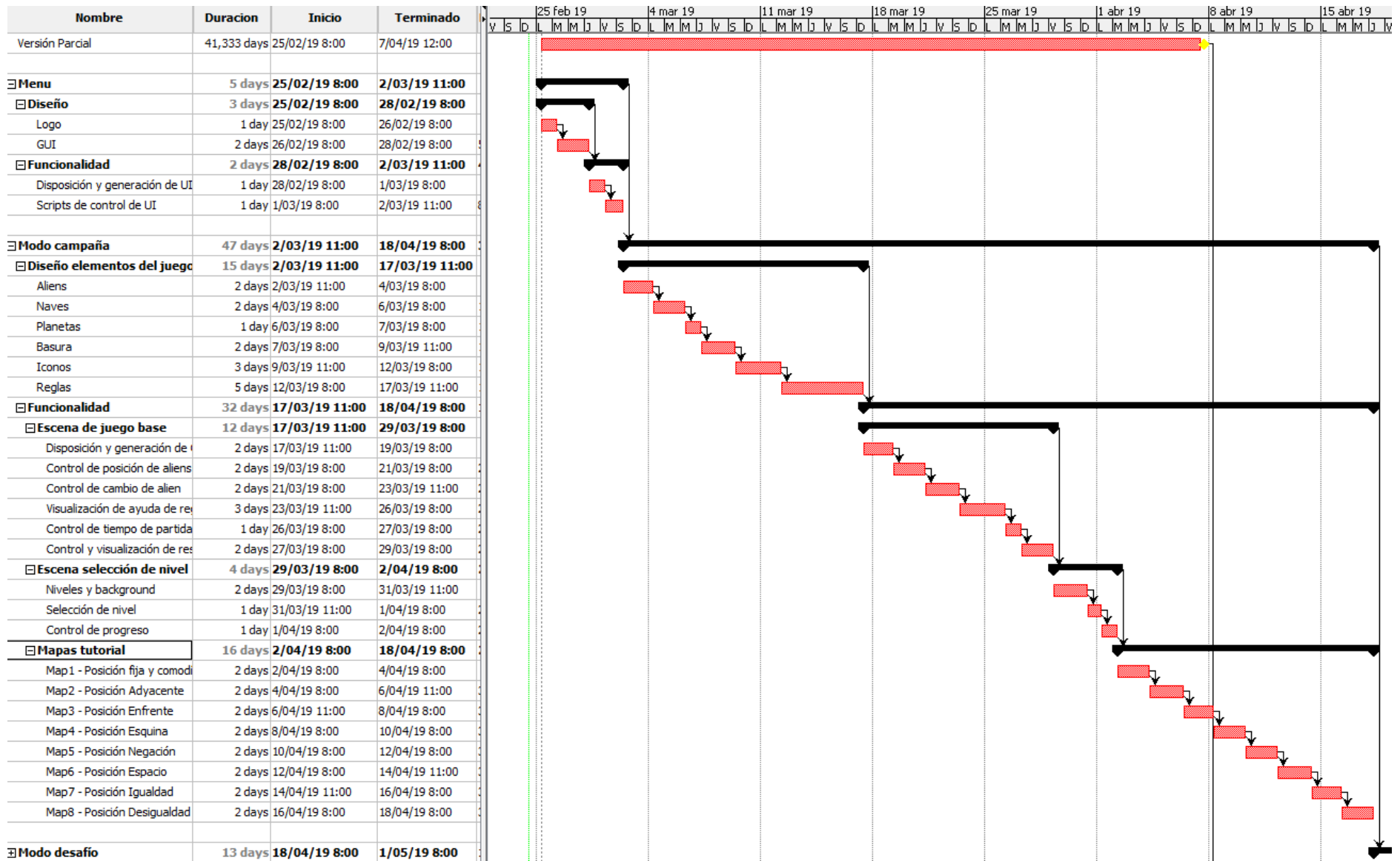


Ilustración 20: Versión Parcial

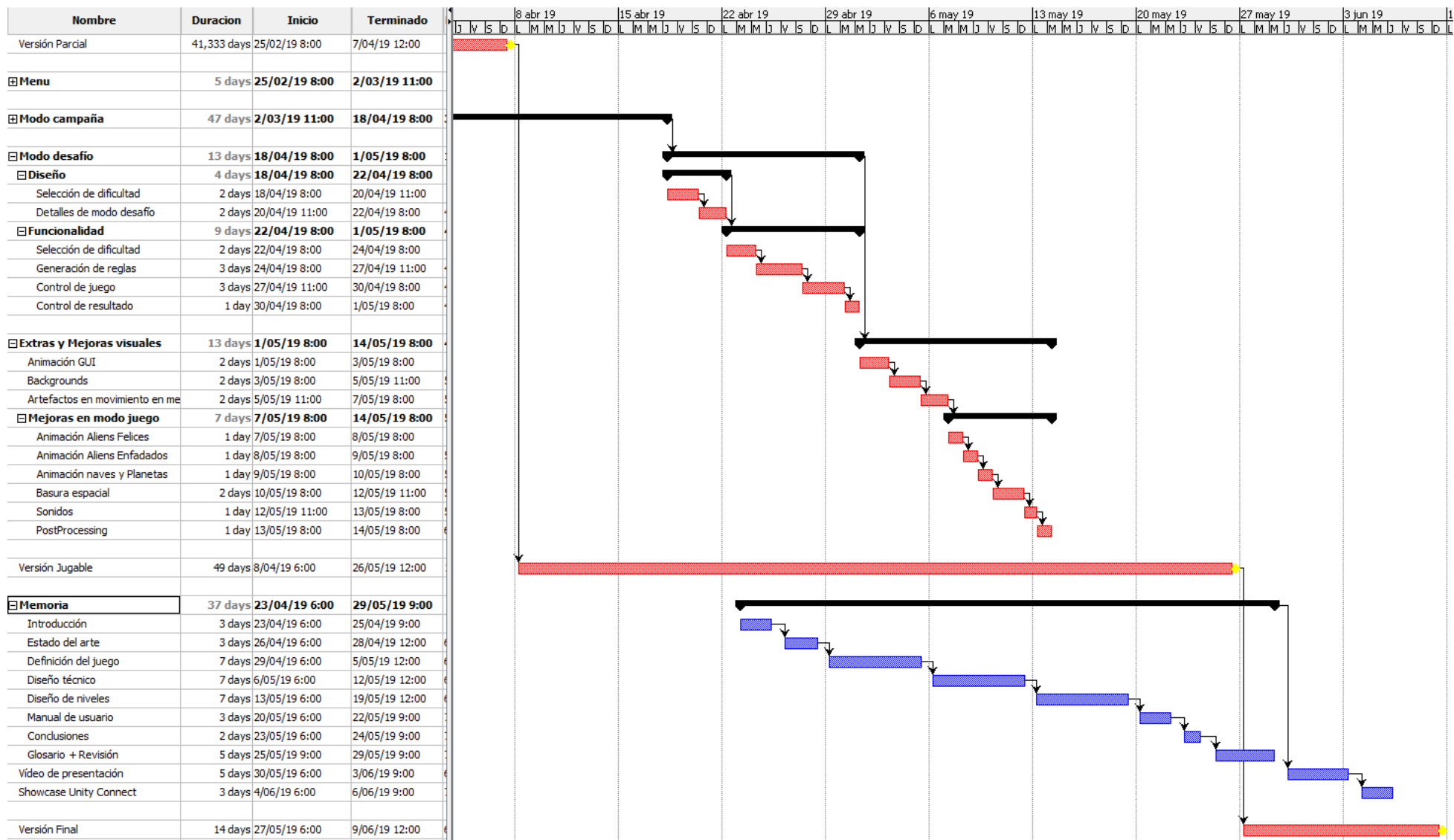


Ilustración 21: Versión Jugable y Entregable Final

4. Diseño técnico

4.2 Entorno de desarrollo

Para el entorno de desarrollo se ha utilizado un equipo con Windows10 en el que se ha instalado todo el software necesario. Además, todo el desarrollo se ha probado en un dispositivo Android Xiaomi Mi Mix 2s.

4.3 Software empleado

Para el desarrollo del proyecto se ha utilizado el siguiente software:

- Unity3d:
 - Unity Hub 2.0.0
 - Unity Version: 2019.3.0a2
- Photoshop CS6 – Administración y procesado de elementos gráficos.
- Krita – Prototipado de elementos gráficos.
- Bosca Ceoil – Diseño de sonidos y música del juego.
- GoldWave – Maquetado y procesado de audio.
- Camtasia – Capturas de pantalla y edición de vídeo.

4.4 Assets

Se han incluido algunos assets oficiales de Unity para la monetización y el análisis de uso, previos a la publicación en el Android Store:

- TextMesh Pro
- Unity Ads
- Unity Analytics

Excepto los assets mencionados, todas las soluciones de código implementadas en el proyecto son propias y no se han incluido assets de ningún tipo de contenido en la mecánica principal. Por tanto, todos los recursos gráficos, sonido, código, etc son propios.

4.5 Arquitectura del juego

Debido al carácter de este documento, no procede una explicación detallada del código generado, no obstante en el siguiente apartado se van a tratar dos puntos destacables:

- En primer lugar, se analizará el flujo de componentes durante una partida en el modo aventura y el modo desafío.
- En segundo lugar, se comentará brevemente las clases y objetos más relevantes y su relación entre sí.

El diagrama de la ilustración 22 muestra el comportamiento general durante la fase de juego en el modo aventura.

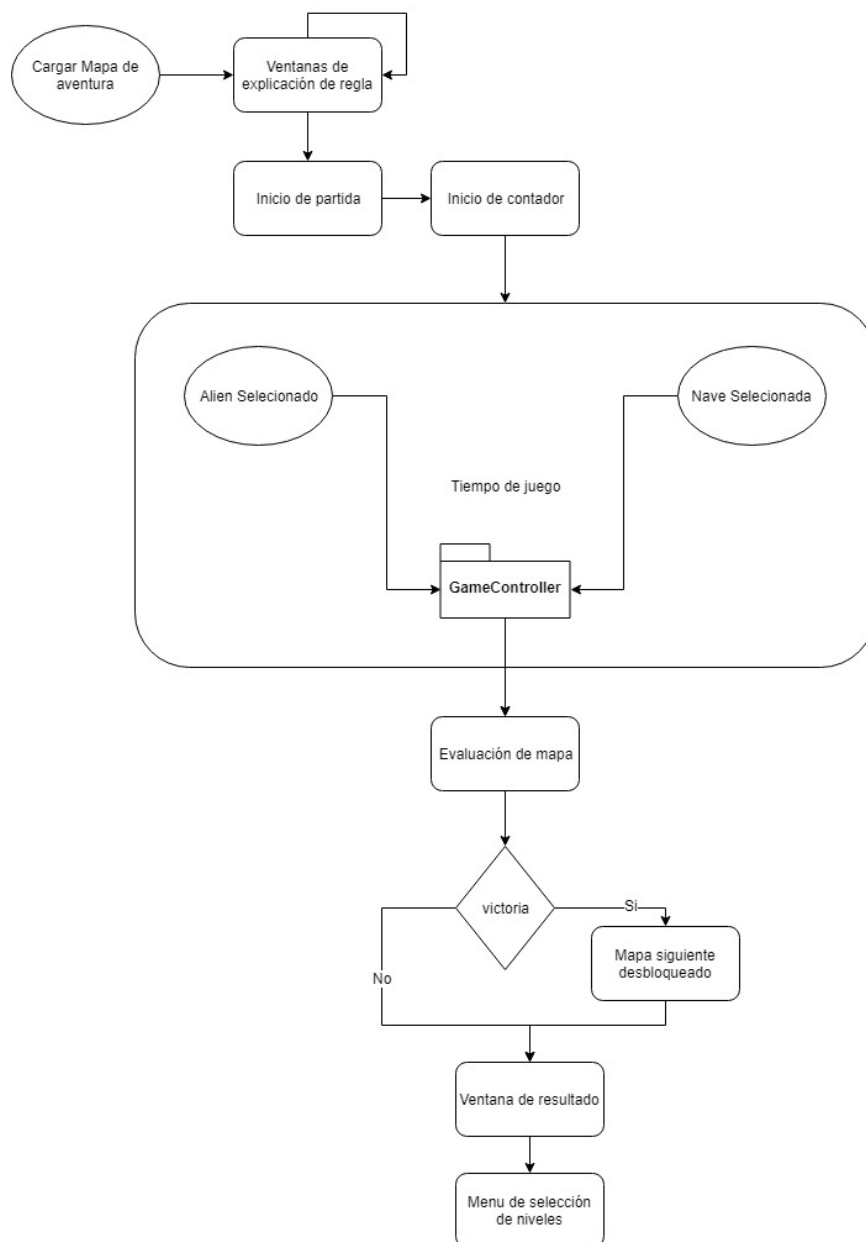


Ilustración 22: Diagrama de flujo - modo aventura

Los mapas del modo aventura contienen un componente MapConfig que tiene preconfiguradas las reglas de cada Alien utilizadas en ese nivel, así como una serie de componentes de ayuda para el tutorial guiado del usuario.

En cambio, en el caso de uso de un mapa del modo desafío, existen diferencias, tal y como se puede ver en el diagrama de la ilustración 23.

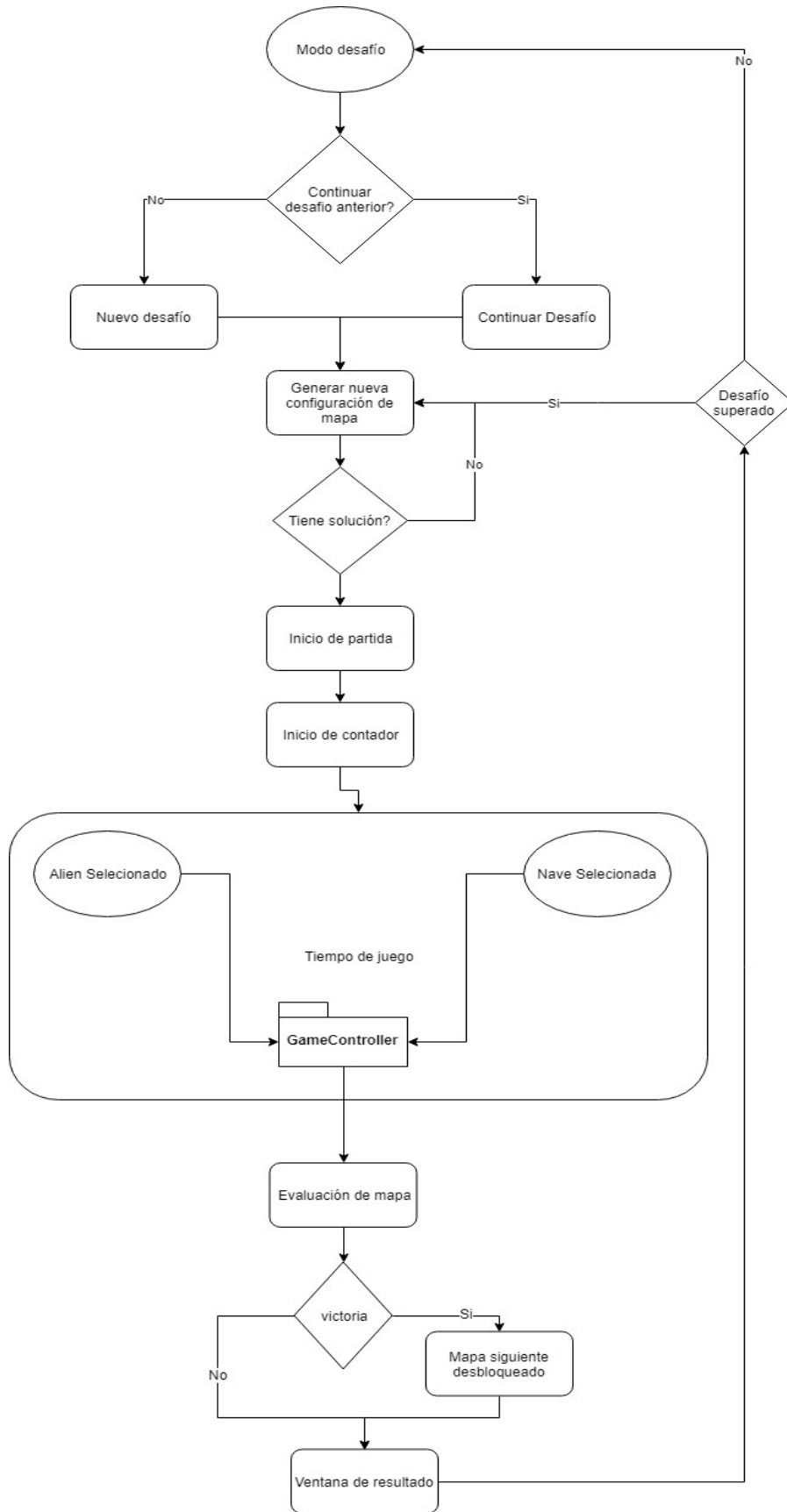


Ilustración 23: Diagrama de flujo modo desafío

Durante el modo desafío, el jugador podrá iniciar un nuevo desafío o retomar el anterior, cargando la puntuación y nivel del último intento. Tras este punto, el juego generará una configuración de mapa válida e iniciará el nivel.

Una vez completado el tiempo de juego, si el mapa se ha resuelto satisfactoriamente, el desafío podrá continuar. Para poder superar el nivel, en el caso del modo desafío no se permiten errores, de lo contrario se devolverá al jugador al menú del juego.

Principales clases y objetos

Dentro del proyecto, en la carpeta Scripts podemos encontrar todos los scripts del juego divididos en tres paquetes:

- **GameLogic**, donde están todos los objetos que manejan la lógica del juego.
- **GuiInterface**, para todos aquellos scripts que interactúan o modifican la interfaz de usuario.
- **Helpers**, para el resto de componentes con carácter de ayuda a los paquetes anteriores.

En esta memoria no se entrará a comentar el código ni su estructura, pero es interesante revisar la aproximación que se ha seguido para la lógica del juego, control de elementos, tablero y sus reglas.

El diagrama incluido en la ilustración 24 muestra la relación entre los principales controladores y clases del juego.

Se ha intentado mantener el principio SOLID durante todo el desarrollo del proyecto, así como la aplicación de patrones de diseño, por ejemplo se han aplicado los patrones : singleton, factory y facade.

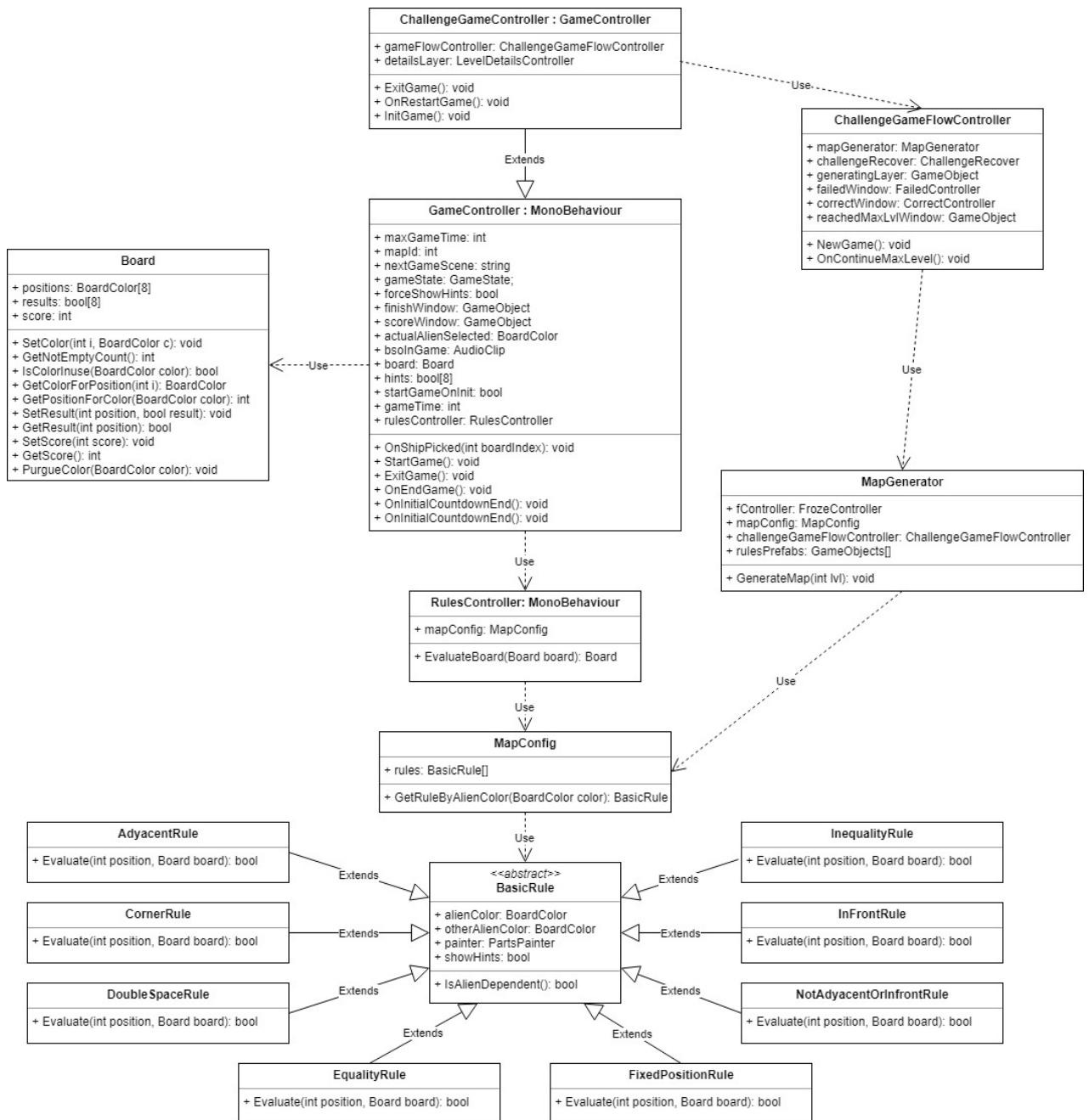


Ilustración 24: SpaceBuddy - UML entidades principales.

4.5 Importación del proyecto

El gitignore del repositorio incluye las reglas de exclusión oficiales para proyectos Unity. Por tanto, las carpetas Library o archivos de solución como sln, apk, etc, no están incluidos en el repositorio.

Para la importación del proyecto es necesario:

- Instalar Unity Hub 2.0.0 - Visual Studio Community 2017 Version: 15.9.6
- Abrir Unity Hub 2.0.0

- Pestaña de Projects -> Add -> seleccionar directorio clon de la rama master del repositorio.
- Pestaña de Installs -> Add -> 2019.3.0a2 para descargar la versión de unity utilizada en el proyecto.
- Pestaña de Projects -> Doble click en SpaceBuddy para iniciar el proyecto y regenerar los archivos temporales

5. Diseño de niveles

Para el diseño de niveles se ha generado una herramienta que permitiese evaluar la dificultad de una configuración de mapa, pudiendo navegar entre las configuraciones existentes y visualizar todas sus posibles soluciones.



Ilustración 25: Evaluador de mapas

“Oculta en el modo desarrollador, el evaluador de mapa permite cargar un prefab de configuración y visualizar todas las soluciones posibles de la configuración de mapa.”

De este modo, la tarea de generar niveles consiste en generar prefabs de MapConfig (mapas de configuración) tal y como se ve en la ilustración 26, medir su complejidad y ajustar según se necesite.

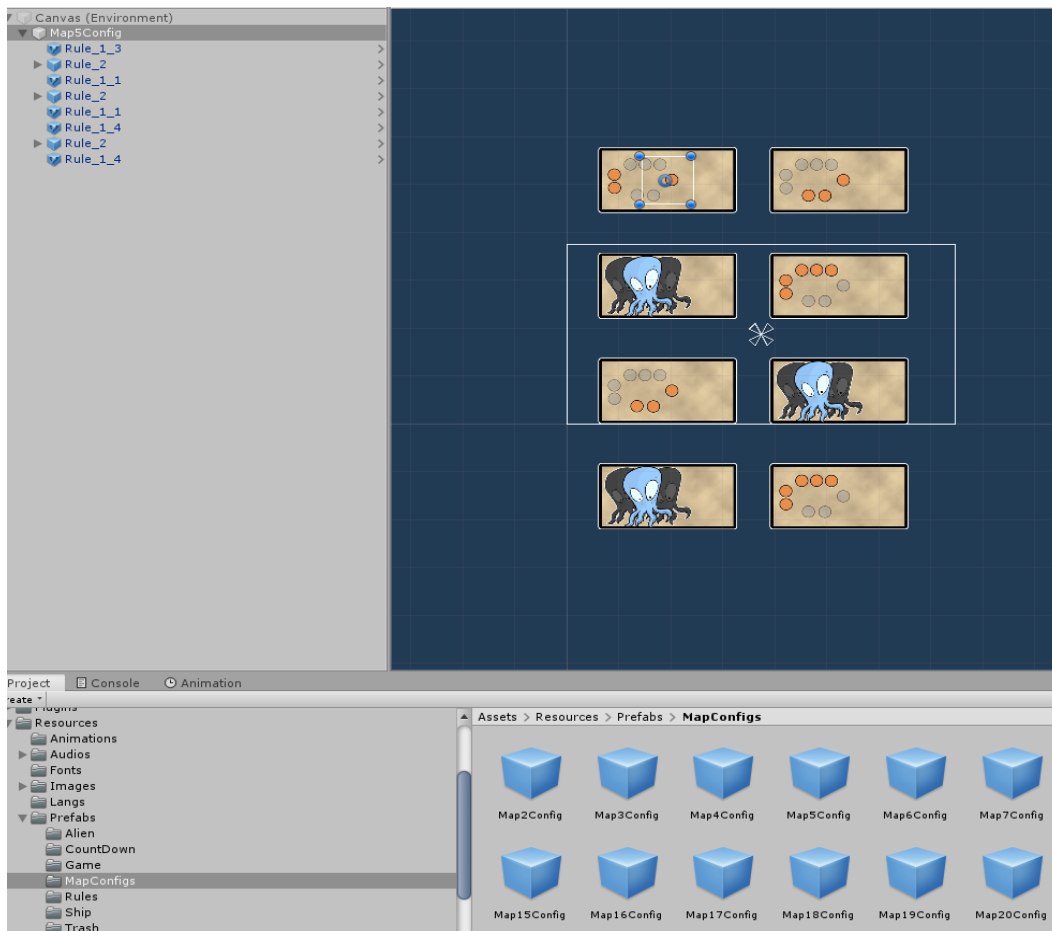


Ilustración 26: Ejemplo configuración de mapa

Por otro lado, adicionalmente se ha creado un objeto `ExtendedTutorialController` para ser incluido en todos los mapas que introducen una mecánica nueva, de forma que sirva de guía en los primeros pasos del jugador durante los primeros segundos de juego con una nueva regla.

Además, durante la fase de configuración de mapa, las reglas pueden ser marcadas para emitir ayudas visuales, de forma que en tiempo de juego se iluminen las posiciones válidas.

6. Evaluación del resultado

Durante la fase beta del juego se envió un compilado del juego a tres personas del círculo de confianza del desarrollador y se solicitó que estas a su vez lo enviaran a otras personas para intentar distribuir el juego entre gente desconocida, intentando compensar así un posible sesgo positivo a la hora de emitir opiniones.

Respecto a los perfiles de los encuestados se pudo destacar dos aspectos:

Edad de usuarios Encuestados

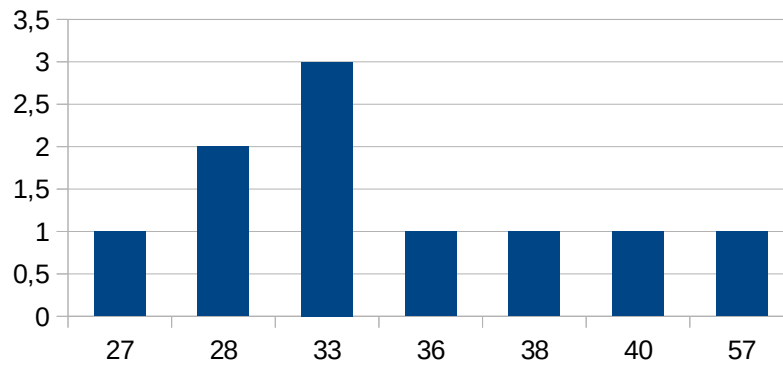


Ilustración 27: Distribución de edad de usuarios encuestados

Nivel de estudios

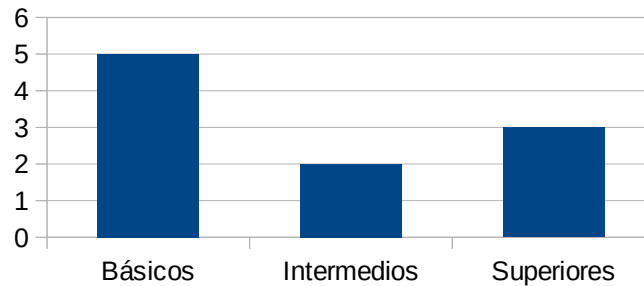


Ilustración 28: Nivel de estudios de usuarios encuestados

La edad de los usuarios encuestados está comprendida entre los 27 y 57 años, Además, existe variedad respecto al nivel de estudios de cada usuario. Por último, la mayoría de los usuarios no conocían el juego de mesa ni el proyecto. Por tanto, se puede considerar que la muestra puede resultar representativa.

A continuación, se hará un repaso de las principales preguntas enviadas a los usuarios del grupo de prueba:

- ¿Has completado el tutorial?
- ¿Te han parecido claras las explicaciones de cada regla?
- ¿Cómo evaluarías el nivel de dificultad de los mapas del tutorial?
- ¿Te parece suficiente el tiempo disponible para resolver el puzzle?
- ¿Te has sentido desorientado por no saber que hacer en algún momento?
- ¿Te han gustado los gráficos y animaciones del juego?
- ¿Te ha gustado la música y sonido del juego?
- Evalúa del 1 al 10 cómo de entretenido te ha parecido el juego.

¿Has completado el tutorial?

10 responses

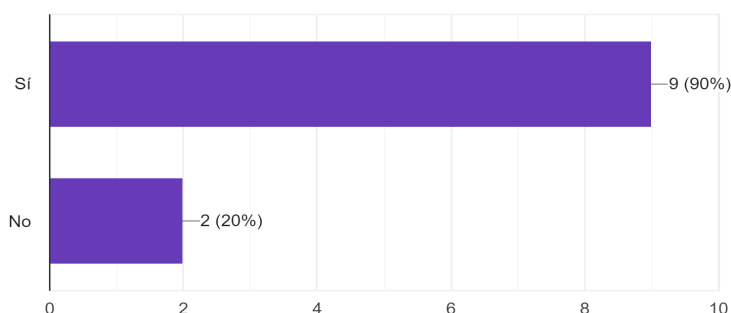


Ilustración 29: Feedback de usuarios - ¿Has completado el tutorial?

En general, se solicitó a todos los usuarios que completasen el modo desafío (en la encuesta se refiere a este modo como tutorial), para así poder evaluar el conjunto de reglas.

Como se puede ver en la ilustración 27, el 90% de los usuarios que realizaron el test habían completado el modo aventura (20 mapas) al emitir su opinión.

¿Te han parecido claras las explicaciones de cada regla?

10 responses

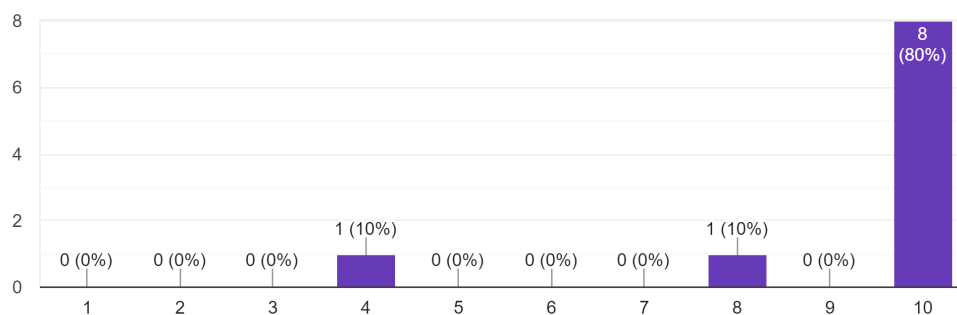


Ilustración 30: Feedback de usuarios - ¿Te han parecido claras las explicaciones de cada regla?

En general, los resultados obtenidos en el test de usuarios muestran que; El esfuerzo invertido en sistemas de ayuda, guía de primeros pasos, asistencia a reglas nuevas y las explicaciones de cada regla, son un entorno suficiente para la comprensión de las mecánicas de cada tipo de Alien.

¿Cómo evaluarías el nivel de dificultad de los mapas del tutorial?

10 respuestas

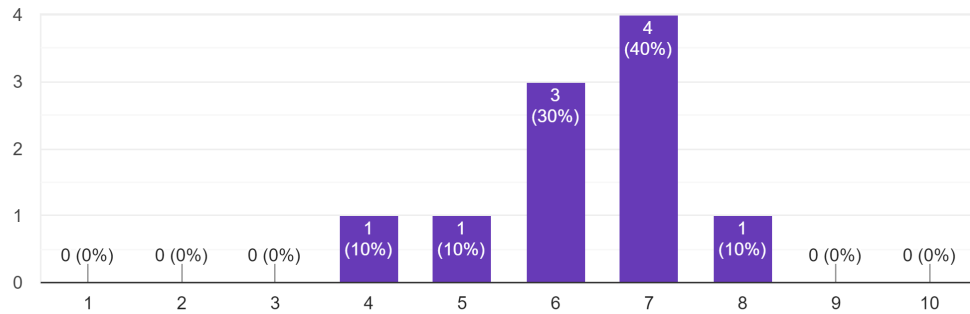


Ilustración 31: Feedback de usuarios - ¿Cómo evaluarías el nivel de dificultad de los mapas?

Los usuarios consideran que la dificultad general del modo aventura es elevada. Estos resultados eran esperados principalmente por dos motivos:

- El núcleo del juego genera puzzles que en esencia son complejos. En función de la combinación de las reglas y cómo de familiarizado esté el usuario con ellas, pueden generarse picos de dificultad mientras asimila una mecánica.
- El modo aventura sintetiza todo el contenido en 20 mapas, por lo que faltan niveles de transición para ayudar al usuario a asentar una mecánica antes de dar paso a una nueva regla.

Por tanto, en un modo aventura con más mapas, la percepción de dificultad sería menor ya que el usuario tendría tiempo de familiarizarse más con una mecánica nueva antes de introducir la siguiente.

¿Te parece suficiente el tiempo disponible para resolver el puzzle?

10 respuestas

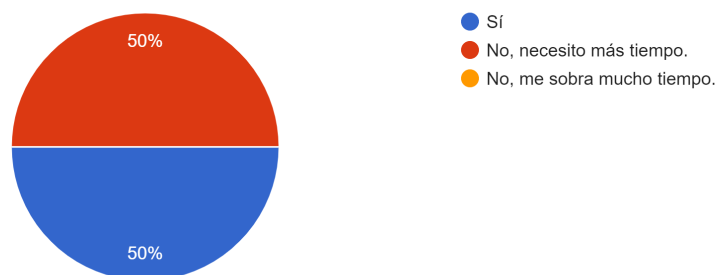


Ilustración 32: Feedback de usuarios - ¿Te parece suficiente el tiempo disponible?

Por lo general, muchos usuarios consideraron que el tiempo de 45 segundos resultaba en la mayoría de los casos insuficiente, y que solo gracias a que los mapas eran estáticos y no variaban, podían conseguir superar el nivel con algunos reintentos llegado a cierto punto de la aventura.

Por ello, se aumentó el tiempo disponible a 60 segundos. Además, durante el tiempo que dura el modo de ayuda o sistema guiado, el tiempo de juego permanece congelado.

¿Te has sentido desorientado por no saber que hacer en algún momento?

10 responses

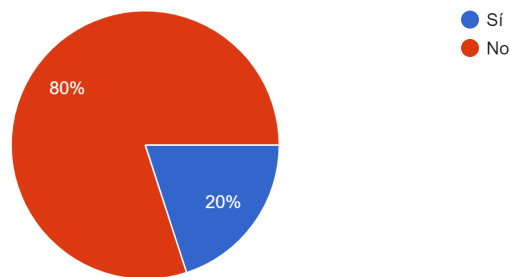


Ilustración 33: Feedback de usuarios - ¿Te has sentido desorientado?

Dos usuarios reportaron sentirse desorientados, y tras analizar los motivos, se tomaron medidas para reducir las dudas que podían surgir. En general se trataban de explicaciones y textos.

¿Te han gustado los gráficos y animaciones del juego?

10 responses

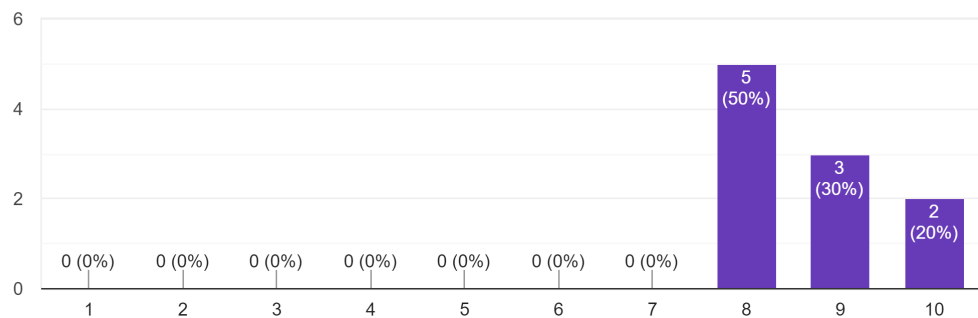


Ilustración 34: Feedback de usuarios - ¿te gustan los gráficos y animaciones?

Respecto a la estética del juego, como se puede ver, los usuarios consideran que encaja con el concepto general del juego y resulta de su agrado.

¿Te ha gustado la música y sonidos del juego?

10 respuestas

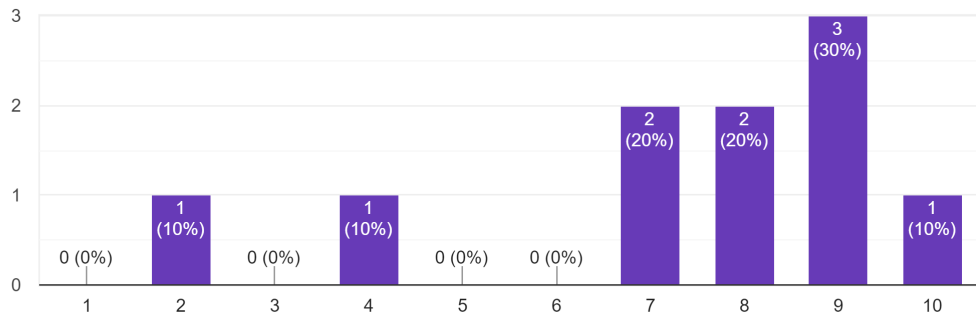


Ilustración 35: Feedback de usuarios - ¿Te gusta la música del juego?

Respecto a la música, no hay tanta unanimidad, aunque por lo general ha resultado del agrado de los jugadores, también se han reportado casos en los que se han obtenido muy bajas puntuaciones.

Evalúa del 1 al 10 como de entretenido te ha parecido el juego.

10 respuestas

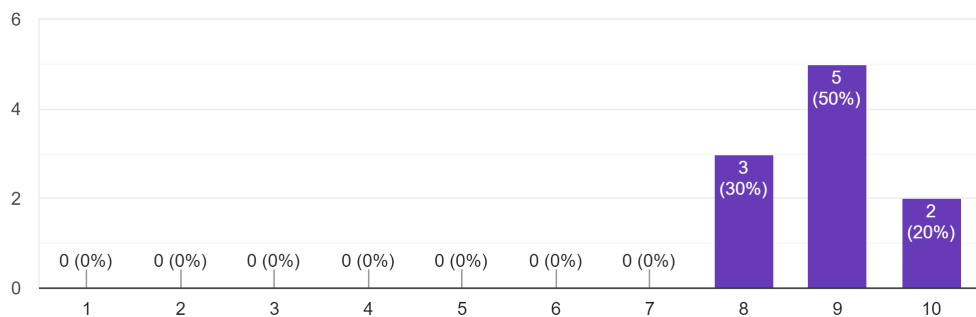


Ilustración 36: Feedback de usuarios - ¿Cómo de entretenido te parece el juego?

Por último, la evaluación general del juego y su mecánica ha obtenido una puntuación muy notoria por lo que se puede deducir que el juego resulta adictivo y entretenido.

7. Manual de usuario

Debido a su extensión, se ha incluido el Manual de usuario como un documento a parte, ver Anexos.

8. Conclusiones

Tras concluir la fase de desarrollo y en perspectiva de todo el periodo y trabajo realizado, me gustaría destacar algunos aspectos:

Unity3D es un motor muy potente y muy intuitivo. Adaptarse a el ha resultado una tarea de menor complejidad de la estimada, además existe gran cantidad de contenido *online* para poder superar cualquier problema.

Por otro lado, durante el desarrollo se identificaron partes técnicas que podrían resultar complejas pero que gracias al *engine* han supuesto muy poco esfuerzo. Por ejemplo, la representación y gestión de las reglas, composición, etc. Por tanto, considero que la elección del *engine* fue un acierto.

Respecto a la planificación inicial, se incluyó cierto margen de seguridad debido al desconocimiento del *engine* y a la falta de practica de ciertas disciplinas tales como: diseño, composición, maquetación, sonido, etc.

Aunque se dedicó más tiempo del estimado, el margen fue excesivo, por lo que hubo que planificar nuevo contenido para extender y mejorar el proyecto. Esto en si mismo es algo positivo puesto que ha permitido generar un producto de mejor calidad, no obstante es un desvío en las estimaciones. Pero como tal pudo detectarse y actuar en consecuencia.

En cualquier caso, se puede concluir que se han alcanzado todos los objetivos planteados durante la fase de diseño y otros adicionales como beneficio del desvío en las estimaciones. En este sentido, el seguimiento de la planificación ha sido un éxito.

Respecto a la metodología de trabajo utilizada, se planteó un desarrollo en cascada a capas, centrando los primeros objetivos en la funcionalidad más básica y mejorando ciertos aspectos no esenciales en etapas posteriores del desarrollo.

Considero que fue una buena decisión ya que permitió cumplir con cada hito entregable, logrando de forma temprana, una primera versión del juego básica con toda la funcionalidad, para posteriormente mejorar el producto añadiendo nuevos sistemas de ayudas, efectos visuales, etc.

De esta forma, se alcanzó una versión jugable rápidamente que fue mejorada en cada ciclo de desarrollo.

Por último, como parte del desarrollo se han incluido algunos complementos pensados para fases futuras del producto, en concreto para la publicación.

Por ello, el juego tiene assets de monetización y control de uso, aunque en la entrega aportada junto a este documento están desactivados para facilitar la evaluación del juego. No obstante, existen algunos cambios necesarios antes de poder publicar el juego para lograr que el juego alcance su máxima calidad.

Por ello, queda pendiente incluir más niveles en el modo aventura, para así generar nuevos niveles de transición entre una regla nueva y la siguiente, consiguiendo que la curva de dificultad sea más progresiva y los jugadores puedan asentar las mecánicas de juego antes de intentar asimilar una nueva.

9. Glosario

- **APK (Android Application Package):** Paquete compilado para el sistema operativo Android.
- **Desarrollo INDIE:** Proyectos de desarrollo sin el apoyo financiero de una distribuidora de videojuegos. Generalmente, se considera que el desarrollo Indie es realizado por pequeños estudios o grupos de personas.
- **Bug:** Error en tiempo de ejecución de una aplicación de software.
- **Gameplay:** Forma en la que el jugador interactúa con el juego o vídeo en el que se reproduce como el jugador interactúa con el juego.
- **GDD (Game Document Design):** Documento que recoge los puntos de especificación y diseño de un videojuego.
- **Script:** Pieza de código
- **Tap:** Interacción en pantalla mediante la pulsación de un elemento visual.
- **Tripe A:** Juego de la máxima calidad.
- **Steam:** Plataforma de publicación de videojuegos para PC.
- **Switch:** Consola portátil y de sobremesa de la compañía Nintendo.
- **GUI:** Interfaz de usuario, capa donde se agrupan los componentes visuales del juego, botones, textos, etc.
- **Android Store:** Tienda virtual de aplicaciones para dispositivos Android.
- **App Store:** Tienda virtual de aplicaciones para dispositivos IOS.
- **C#:** C sharp, lenguaje de programación utilizado en Unity3D.
- **Alpha:** Fase temprana del software, primera versión ejecutable.
- **Beta:** Fase completa del software, todavía en revisión y ejecución.
- **Release Candidate:** Primera versión del ejecutable, software terminado.
- **Asset:** Conjunto de objetos de Unity3d que añaden nuevo contenido al proyecto.

- **Prefab:** Objeto de Unity3d que encapsula e independiza contenido para su uso posterior.
- **Trigger:** Evento producido durante la ejecución de software para su evaluación.
- **Diagrama de gantt:** Diagrama que muestra la consecución de objetivos en una línea temporal.
- **Preset:** Configuración preestablecida.
- **MIDI:** Formato de archivo de audio que almacena información de entrada para el tratamiento digital de instrumentos.
- **Bosca Ceoil:** Software de composición musical mediante patrones de notas utilizando presets MIDI de diferentes instrumentos.
- **Asset Store** Tienda Unity3D en la que se pueden encontrar elementos utilizables en el desarrollo: Modelos gráficos, sonido, piezas de código, etc.

10. Bibliografía

- A Bronze Age game called 58 holes was found chiseled into stone in Azerbaijan - 16 Noviembre 2018 - <https://www.sciencenews.org/article/bronze-age-game-found-chiseled-stone-azerbaijan>

11. Anexos

- Manual Usuario – SpaceBuddy
- APK SpaceBuddy Release candidate 1