



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

Desarrollo de un videojuego en Unity:  
implementación del *back-end*

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor:** Marc Garcia Ferrer

**Tutor:** Francisco José Abad Cerdá

2020-2021



# Resumen

---

Este proyecto tiene como objetivo implementar un videojuego en el entorno de desarrollo Unity. El videojuego se ha realizado con un enfoque de emprendimiento para una posible futura salida al mercado. Su desarrollo ha seguido una metodología ágil. El TFG abarca desde la concepción de la idea de negocio hasta la finalización del segundo MVP. El proyecto se ha realizado dentro del marco de Start.inf.

El proyecto, con título *Blitz Party*, es un videojuego multijugador para dispositivos móviles que pertenece al género de los juegos basados en minijuegos. Se trata de una aplicación *free-to-play* que dispone de contenido extra adquirible mediante micropagos. Consta también de funciones sin conexión y personalización de personajes, entre otros. El proyecto se ha desarrollado por un equipo compuesto por cuatro alumnos, en el marco del desarrollo de sus TFG, donde cada uno se ha centrado en un área del proyecto.

Este trabajo se enfoca hacia la implementación del *back-end*, es decir, la parte del desarrollo que es invisible para los usuarios del videojuego. Concretamente, el *back-end* consiste en la toma, guardado y recuperación de datos y en la lógica que procesa las acciones de los usuarios y devuelve una salida al *front-end* para mostrarla a los jugadores. Para esto se usan diversas herramientas y técnicas, que están especializadas en realizar las tareas mencionadas.

**Palabras clave:** videojuego, Unity, minijuegos, emprendimiento, *back-end*.

# Resum

---

Aquest projecte té com a objectiu implementació un videojoc en l'entorn de desenvolupament Unity. El videojoc s'ha realitzat amb un enfocament d'emprenedoria per a una possible eixida al mercat. El seu desenvolupament ha seguit una metodologia àgil. El TFG comprén des de la concepció de la idea de negoci fins l'acabament del segon MVP. El projecte s'ha realitzat dins del marc de Start.inf.

El projecte, amb títol *Blitz Party*, és un videojoc multi jugador per a dispositius mòbils que pertany al gènere dels jocs basats en mini jocs. Es tracta d'una aplicació *free-to-play* que compta amb contingut addicional adquirible mitjançant micropagaments. Compta també amb funcions sense connexió i personalització de personatges, entre altres. El projecte s'ha desenvolupat per un equip



format per quatre alumnes, en el marc del desenvolupament dels seus TFG, on cada un s'ha centrat en una àrea del projecte.

Aquest treball s'enfoca cap a la implementació del *back-end*, és a dir, la part del desenvolupament que és invisible per als usuaris del videojoc. Concretament, el *back-end* consisteix en l'agafament, guardat i recuperació de dades i en la lògica que processa les accions dels usuaris i torna una eixida al *front-end* per a mostrar-la als jugadors. Per açò s'utilitzen diverses ferramentes i tècniques, que estan especialitzades en realitzar les feines mencionades.

**Paraules clau:** videojoc, Unity, mini jocs, empenedoria, *back-end*.

## Abstract

---

This project aims to create a videogame using the Unity Game Engine. The video game has been made with an entrepreneurial approach with a projection for a possible future release to the market. Its development has followed an agile methodology. This end-of-degree project spans from the conception of the business idea until the ending of the second MVP. The project has been made within the framework of Start.inf.

The project, titled Blitz Party, is a multiplayer video game for mobile devices that belongs to the genre of games based on minigames. It is a free-to-play application that includes extra content that can be purchased with microtransactions. It also contains offline features and character customization, among many others. This project has been developed by a team of four students, within the framework of the development of their TFG, where each one has focused on one area of the project.

The main focus of this work is the implementation of the back-end, which is the part of the development that is not visible to the users of the videogame. Concretely, the back-end consists on the take, storage and retrieval of data and the logic which processes the actions of users and returns an output to the front-end to be shown to the players. In order to do this, several tools and techniques that are specialized in carrying out the aforementioned tasks were used.

**Keywords:** video game, Unity, minigames, entrepreneurial, back-end.

# Tabla de contenidos

---

1. Introducción	12
1.1 Motivación	13
1.2 Justificación del tema	14
1.3 Objetivos	14
1.4 Estructura de la memoria	15
2. Evaluación de la idea de negocio	16
2.1 Estado del arte	16
2.2 Estudio de mercado	19
2.2.1 Super Mario Party	20
2.2.2 Wario Ware	23
2.2.3 Among Us	25
2.2.4 Dumb Ways to Die	27
2.3 Análisis DAFO	29
2.4 Modelo de negocio y proyección económica	30
2.5 Lean Canvas	36
2.6 Conclusiones	39
3. Desarrollo de la idea de negocio	40
3.1 Mapa de características	41
3.2 Primer <i>Minimum Viable Product</i>	42
3.2.1 Desarrollo del <i>MVP</i>	43
3.2.2 Experimento	45
3.3 Segundo <i>Minimum Viable Product</i>	53
3.3.1 Desarrollo del <i>MVP</i>	54
3.3.2 Experimento	55
3.4 Publicación del producto y <i>marketing</i>	58
4. Tecnologías utilizadas	60



4.1 Unity Engine	60
4.2 Visual Studio	60
4.3 Git/GitHub	61
4.4 Worki Process	62
4.5 Photoshop	62
4.6 Mirror	63
4.7 Azure PlayFab	63
5. Desarrollo del <i>back-end</i> : Análisis	65
5.1 Introducción	65
5.2 Métodos usados	65
5.3 Brainstorming	66
5.4 Casos de uso	68
6. Desarrollo del <i>back-end</i> : Diseño	71
6.1 Introducción	71
6.2 Patrones	72
6.3 Diagrama de clases	73
7. Desarrollo del <i>back-end</i> : Implementación	74
7.1 Implementación del <i>back-end</i> local	74
7.2 Implementación del <i>back-end</i> en la nube	76
8. Desarrollo del <i>back-end</i> : Resultados	81
8.1 Introducción	81
8.2 Pruebas de aceptación	81
8.3 Encuestas a usuarios	83
9. Trabajo adicional	85
9.1 Pathfinding en Binky Pursuit	85
9.2 Aparición de enemigos en Whack-A-Mole	87
9.3 Animaciones en Cowboy Duel	88
10. Conclusiones	92
11. Trabajo futuro	93

12. Apéndices	94
12.1 Listado de abreviaturas, siglas y acrónimos	94
12.2 Lista de recursos gráficos	94
12.3 <i>Game Design Document</i> (GDD)	96
13. Referencias bibliográficas	121



# Listado de figuras

---

Figura 1. Logo de Blitz Party .....	12
Figura 2. Enlace QR de descarga de Blitz Party .....	13
Figura 3. Lanzamiento de jabalina en Olympic Decathlon .....	16
Figura 4. Partida de Bomberman Party Edition .....	17
Figura 5. Personaje personalizable de la Nintendo Wii .....	18
Figura 6. Captura del videojuego Buzz!: el Gran Reto .....	18
Figura 7. Cronología juegos party o de minijuegos .....	19
Figura 8. Tabla comparativa de productos competidores.....	20
Figura 9. Portada de Super Mario Party .....	21
Figura 10. Modo tablero de Super Mario Party.....	21
Figura 11. Minijuego de Super Mario Party.....	22
Figura 12. Portada de Wario Ware Gold.....	23
Figura 13. Microjuego de Wario Ware Gold .....	24
Figura 14. Cinemática del modo historia de Wario Ware Gold .....	24
Figura 15. Portada de Among Us .....	25
Figura 16. Impostor asesinando a un tripulante en Among Us .....	26
Figura 17. Jugador realizando una tarea en Among Us.....	27
Figura 18. Portada de Dumb Ways to Die.....	28
Figura 19. Fotograma de la canción Dumb Ways to Die .....	28
Figura 20. Microjuego de Dumb Ways to Die .....	29
Figura 21. Matriz DAFO de Blitz Party .....	30
Figura 22. Proyección económica año 1 .....	32
Figura 23. Proyección económica año 2 .....	33
Figura 24. Personal contratado.....	33
Figura 25. Proyección económica años 3 y 4.....	34
Figura 26. Número de descargas año 5 .....	35
Figura 27. Resultado acumulado de ingresos.....	36
Figura 28. Lean Canvas de Blitz Party.....	38
Figura 29. Pitch Doc.....	41
Figura 30. Mapa de características de Blitz Party .....	42
Figura 31. Mapa de características del primer MVP.....	43
Figura 32. Minijuegos seleccionados del primer MVP.....	43
Figura 33. Captura del minijuego Binky Pursuit.....	44
Figura 34. Captura del minijuego Whack-A-Mole. ....	44

Figura 35. Captura del minijuego Cowboy Duel. ....	45
Figura 36. Tuit de promoción del primer MVP.....	46
Figura 37. Edad de los participantes de la encuesta .....	47
Figura 38. Tiempo que dedican a videojuegos los participantes de la encuesta .....	47
Figura 39. Porcentaje de participantes que juegan en dispositivos móviles.....	47
Figura 40. Horas que juegan los participantes de la encuesta en dispositivos móviles.....	48
Figura 41. Minijuego favorito de los encuestados .....	48
Figura 42. Puntuaciones de la curva de aprendizaje.....	49
Figura 43. Puntuaciones sobre la dificultad del juego.....	49
Figura 44. Opinión sobre dificultad excesiva de algún minijuego .....	50
Figura 45. Opiniones sobre la claridad de la interfaz gráfica.....	50
Figura 46. Opiniones sobre diseño y ambientación .....	51
Figura 47. Opinión sobre la música de los minijuegos .....	51
Figura 48. Puntuaciones generales de Blitz Party .....	52
Figura 49. Mapa de características del segundo MVP .....	53
Figura 50. Binky Pursuit multijugador.....	54
Figura 51. Whack-A-Mole multijugador.....	55
Figura 52. Pregunta rango de edad.....	56
Figura 53. Pregunta jugadores de dispositivos móviles .....	56
Figura 54. Pregunta horas jugadas en dispositivos móviles .....	57
Figura 55. Pregunta minijuego preferido .....	57
Figura 56. Pregunta puntuación general Blitz Party.....	58
Figura 57. Blitz Party en la Play Store .....	58
Figura 58. Países desde los que se ha descargado Blitz Party .....	59
Figura 59. Cuenta de Twitter de Swifter Games .....	59
Figura 60. Logo de Unity Engine.....	60
Figura 61. Logo del IDE Visual Studio.....	60
Figura 62. Logos de Git y GitHub respectivamente.....	61
Figura 63. Logo de Worki TUNE-UP Process.....	62
Figura 64. Logo de Adobe Photoshop.....	62
Figura 65. Logo de Mirror.....	63
Figura 66. Logo de Azure PlayFab .....	64
Figura 67. Brainstorming para Blitz Party .....	67
Figura 68. Caso de uso Registro de usuario .....	68
Figura 69. Caso de uso Inicio de sesión de usuario.....	68
Figura 70. Caso de uso Ganar acorns .....	69
Figura 71. Caso de uso Guardar volumen de música o efectos de sonido .....	69



Figura 72. Caso de uso Comprar objetos de la tienda .....	69
Figura 73. Caso de uso Personalizar personaje .....	70
Figura 74. Esquema de funcionamiento del patrón Fachada.....	72
Figura 75. Diagrama de clases de Blitz Party .....	73
Figura 76. Métodos estáticos de la clase PlayerPrefs (1) .....	74
Figura 77. Métodos estáticos de la clase PlayerPrefs (2) .....	74
Figura 78. Pantalla de inicio de sesión offline .....	75
Figura 79. Importación del paquete de Azure PlayFab .....	76
Figura 80. Editor de Unity tras la instalación del paquete PlayFab .....	77
Figura 81. Editor de Unity tras instalar el SDK de PlayFab .....	77
Figura 82. Pantalla de inicio de sesión online .....	78
Figura 83. Menú de ajustes con cierre de sesión.....	80
Figura 84. Prueba de aceptación para Guardado local de datos sobre el usuario.....	81
Figura 85. Prueba de aceptación para User profile .....	82
Figura 86. Pruebas de aceptación para Online login and Register Functionality.....	82
Figura 87. Pruebas de aceptación para Data Persistence on the Cloud .....	82
Figura 88. Pregunta sobre las estadísticas que se guardan .....	83
Figura 89. Pregunta sobre inicio de sesión con cuentas de terceros.....	84
Figura 90. Objeto pathfinder de Unity. ....	85
Figura 91. Prefab de binky para Binky Pursuit. ....	86
Figura 92. Movimiento de binkies en Binky Pursuit.....	87
Figura 93. Mapa de Whack-a-Mole .....	87
Figura 94. Spawner de topos .....	88
Figura 95. Animación Idle de Cowboy Duel .....	89
Figura 96. Animación Run de Cowboy Duel .....	89
Figura 97. Animación Wait to shoot de Cowboy Duel .....	89
Figura 98. Animación shoot de Cowboy Duel.....	90
Figura 99. Animación death de Cowboy Duel.....	90
Figura 100. Máquina de estados de Cowboy Duel.....	90
Figura 101. Personajes de Blitz Party .....	104
Figura 102. Mockup menú principal Blitz Party .....	106
Figura 103. Pantallas de juego .....	107
Figura 104. Herramientas de los minijuegos.....	107
Figura 105. Escenario de Binky Pursuit.....	109
Figura 106. Interfaz de Binky Pursuit .....	110
Figura 107. Zorros de Binky Pursuit.....	110
Figura 108. Binkies de Binky Pursuit .....	110

Figura 109. Escenario de Whack-A-Mole.....	112
Figura 110. Interfaz de Whack-A-Mole .....	112
Figura 111. Martillo de Whack-A-Mole .....	112
Figura 112. Topo normal de Whack-A-Mole.....	113
Figura 113. Topo dorado de Whack-A-Mole.....	113
Figura 114. Zoomy de Whack-A-Mole.....	113
Figura 115. Escenario de Cowboy Duel.....	115
Figura 116. Interfaz de Cowboy Duel.....	115
Figura 117. Personajes de Cowboy Duel .....	116
Figura 118. Calendario de trabajo .....	119



# 1. Introducción

---

Hoy en día, la industria de los videojuegos está en el punto más álgido de su historia. Los videojuegos se usan como afición, como ocupación profesional en el caso de los *e-sports* o como método para abstraerse de la vida diaria.

Es este último aspecto en el que se enfoca este proyecto. Este trabajo de fin de grado se centra en el desarrollo de un juego multijugador casual para dispositivos móviles. El juego desarrollado, *Blitz Party*, es un juego basado en minijuegos que consisten en pequeños fragmentos jugables de una duración aproximada de un minuto. Los minijuegos tienen un principio y un final por sí mismos y cada uno tiene una temática propia. En la Figura 1 se puede observar el logo de *Blitz Party*, que se utiliza como icono de la aplicación una vez instalada en el dispositivo móvil.



Figura 1. Logo de *Blitz Party*

El videojuego ha sido desarrollado como un proyecto de emprendimiento en el marco de Start.inf, espacio de emprendimiento de la ETSINF coordinado por Patricio Orlando Letelier Torres. Actualmente el juego ha conseguido 74 descargas en la *Google Play Store*. Se puede jugar desde cualquier dispositivo móvil con el sistema operativo *Android* escaneando el código QR de la Figura 2.



Figura 2. Enlace QR de descarga de *Blitz Party*

Este proyecto se compone de los siguientes integrantes, todos ellos estudiantes del último curso del Grado en Ingeniería Informática.

- Alejandro Vicent Micó: desarrollo del apartado *front-end* del videojuego.
- Jose Francisco Gómez Alemany: desarrollo de las mecánicas de juego del videojuego.
- Marc Garcia Ferrer: desarrollo del apartado *back-end* del videojuego, descrito en esta memoria.
- Pablo Stemmer Fernández: desarrollo de las funcionalidades multijugador del videojuego.

Dada la naturaleza de este proyecto, esta memoria se compone de una sección común formada por los capítulos del uno al cuatro, seguida de otra específica de cada uno de los integrantes del equipo en la que se ahondará en el tema de cada miembro.

## 1.1 Motivación

La motivación principal de este proyecto es llevar a cabo el proceso de emprendimiento para sacar un producto al mercado, en este caso un videojuego. Para este equipo de desarrollo, los videojuegos han estado presentes en nuestra vida desde una edad temprana y el poder experimentar el proceso de desarrollo de uno desde la concepción de la idea de negocio hasta la salida al mercado es una gran oportunidad.

Los ingresos del mercado de videojuegos para dispositivos móviles constituyen una gran parte del total del mercado general de videojuegos. En 2019, el mercado ingresó 120,1 mil millones de dólares, 64,4 mil millones de los cuales provienen de los videojuegos para móviles [1]. Debido a esto, se ha decidido que el desarrollo de este videojuego era más viable y se podía conseguir una rentabilidad mayor si se orientaba a dispositivos móviles.

El motivo de la formación del equipo de desarrollo es que debido a la conexión previa entre sus integrantes y a la compenetración existente, se facilita la aplicación de metodologías ágiles. Además, la conexión personal de cada uno con los videojuegos es objeto de motivación para la creación de un producto de calidad y del que sentirse orgulloso.

Por último, la posibilidad de interactuar con los futuros consumidores del producto por medio de la realización de dos MVP y sendos experimentos e implementar el *feedback* que se recibe por medio de encuestas es una oportunidad nueva e ilusionante para los miembros del equipo.

## 1.2 Justificación del tema

Se ha elegido el género de los minijuegos porque durante una sesión de juego de los cuatro miembros del equipo de desarrollo, se encontró que es un género que destaca por la diversión que ofrece, sobre todo entre amigos, y por lo poco explorado que ha sido en el mercado móvil.

Con respecto al estilo artístico, se ha optado por elegir un estilo casual que se adecua a la naturaleza de los propios minijuegos y que además aporta una frescura que busca ayudar a los jugadores a desconectar del estrés diario.

## 1.3 Objetivos

El objetivo principal de este TFG es el desarrollo de un videojuego multijugador para dispositivos móviles, *Blitz Party*.

En cuanto a objetivos específicos, se han establecido los siguientes:

- Realizar dos *minimum viable product* o MVP con sus experimentos correspondientes, con la finalidad de obtener un producto completamente funcional y conseguir *feedback* de los jugadores.
- Indagar en el mundo del emprendimiento, llevando a cabo un proyecto de emprendimiento de principio a fin y utilizando metodologías ágiles para organizarlo y estructurarlo.
- Publicar el videojuego en la tienda digital *Play Store* de *Android* para facilitar la distribución y poder acceder a un mercado mayor.
- Desarrollar la parte *back-end* del videojuego elaborando un sistema de guardado de información de jugadores *online*.

## 1.4 Estructura de la memoria

A lo largo de la memoria se presentan diferentes secciones, empezando por la evaluación de la idea de negocio en la que se explica el estudio de mercado y de la competencia que se ha realizado, un análisis DAFO del producto, un modelo de negocio y su proyección económica para los próximos cinco años, el *Lean Canvas* del producto y por último las conclusiones derivadas del estudio realizado.

En la siguiente sección se desarrolla la propia idea de negocio. En ella se evalúan ambos MVP y cómo han ido evolucionando, junto con el experimento llevado a cabo para cada uno de ellos. Se indaga también en el impacto que tienen las redes sociales en el *marketing* y qué datos se pueden extraer de los usuarios que han descargado el videojuego en sus dispositivos móviles a partir de su experiencia de juego, así como de encuestas realizadas.

A continuación, se exponen las tecnologías utilizadas para el desarrollo del producto, así como el porqué de su elección y su impacto en el mismo. En las secciones siguientes se desarrolla la parte individual del proyecto, que en este caso consiste en la implementación de la parte *back-end*. Esta parte se compone de cuatro secciones diferentes. La primera es la de análisis, en la que se habla de la identificación de los requisitos. La siguiente es la de diseño, en la que se explica el esquema de la solución decidida y la implementación de patrones. En la sección que precede, se habla sobre cómo se ha implementado la solución y en la última de estas cuatro secciones, se explicarán las pruebas efectuadas y los resultados obtenidos. En la parte siguiente a estas se habla sobre el trabajo adicional que se ha efectuado en otras partes del proyecto no pertenecientes al *back-end*.

Por último, pero no menos importante, en las últimas secciones se explica qué conclusiones se extraen del desarrollo del producto, su proyección futura y el trabajo a realizar. A continuación, los apéndices con los acrónimos y siglas utilizadas, la lista de recursos gráficos y el GDD del videojuego. Finalmente, se presenta una bibliografía con las referencias usadas.



## 2. Evaluación de la idea de negocio

El objetivo de esta sección es evaluar la idea de negocio por medio de un estudio del mercado y de los competidores, además de un análisis DAFO, un modelo de negocio, una proyección económica a 5 años y un *Lean Canvas*. Así, se extraen conclusiones sobre la viabilidad del proyecto en el mercado.

### 2.1 Estado del arte

Los juegos del género *party* llevan existiendo desde el año 1980, momento en el que se estrenó el que se considera el primer juego *party* de la historia. Se trata de *Olympic Decathlon*, videojuego desarrollado por *Timothy W. Smith* para TRS-80 y publicado por *Microsoft*. El juego consiste en diez competiciones de atletismo variadas, como por ejemplo lanzamiento de jabalina, visible en la Figura 3. En 1981 fue llevado a *Apple II* y en 1982 se le cambió el nombre a *Microsoft Decathlon* y se estrenó en IBM PC. Se le considera un juego *party* dado que pueden jugar hasta seis jugadores. El modo multijugador consiste en una suerte de competición olímpica en la que los jugadores se turnan para conseguir marcas elevadas y se lleva la victoria el que obtiene la mejor puntuación.

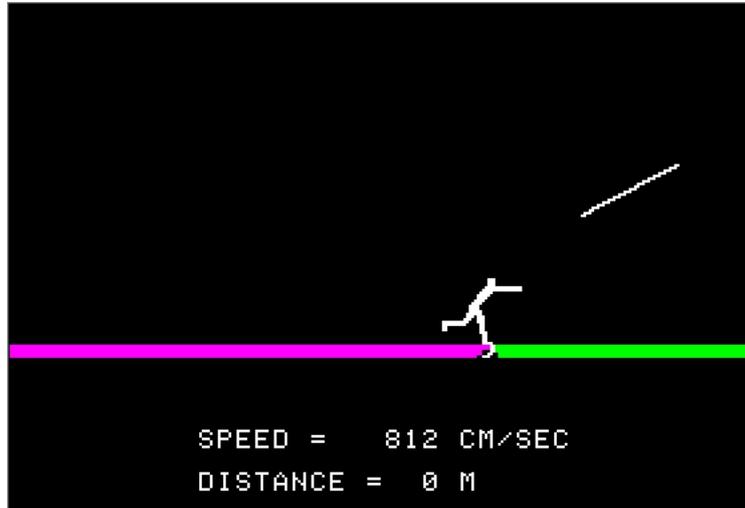


Figura 3. Lanzamiento de jabalina en *Olympic Decathlon*

El género no recibió entregas destacables hasta finales de los años 90, momento en el que salieron al mercado algunas videoconsolas más potentes como la *Nintendo 64* y la *Playstation*. Debido a esto, en el año 1998 se estrenaron dos juegos que definieron el futuro del género *party*, se trata de *Mario Party* y *Bomberman Party Edition*. *Mario Party* sentó las bases de lo que consideramos hoy en día un juego *party*. Este juego consiste en un tablero donde los jugadores tendrán que conseguir el mayor número de estrellas posibles. Para conseguir estas estrellas, deberán vencer a

los otros jugadores en uno de más de 50 minijuegos. El éxito de este juego llevó al estreno de 15 juegos más en la saga, incluyendo *Super Mario Party* para *Nintendo Switch*, juego que se analiza más profundamente en el estudio de mercado.

*Bomberman Party Edition*, al contrario que *Mario Party*, no está basado en minijuegos. Este juego, como se aprecia en la Figura 4, se basa en derrotar al resto de jugadores avanzando por un laberinto y alcanzarlos con la explosión de la bomba que se deposita en el suelo. La fórmula funcionó y hoy en día la saga *Bomberman* goza de una gran popularidad y posee más de 70 juegos.

Se considera un juego *party* porque está orientado al multijugador local. Además, el videojuego da pie a situaciones donde los jugadores pueden divertirse más estando juntos en el mismo sitio. Por ejemplo, poniendo bombas a otros jugadores o jugando una partida con la puntuación ajustada cerca del final.



Figura 4. Partida de *Bomberman Party Edition*

A partir de 2006, con la llegada de la *Nintendo Wii* y sus controles novedosos e intuitivos, se captó a un público más casual, que utilizaba la consola de una manera más social, jugando con amigos. Esto supuso un aumento de la popularidad de los juegos *party*. Un juego que explotó los controles de la *Nintendo Wii* y su aspecto social fue el *Wii Party*, que tenía muchos componentes similares al *Mario Party* y permitía personalizar a los personajes, como se observa en la Figura 5.

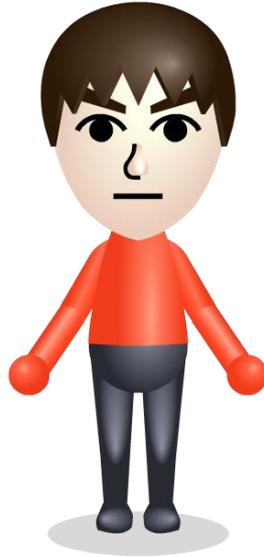


Figura 5. Personaje personalizable de la *Nintendo Wii*

Por otra parte, Sony también se sumó a esta corriente de juegos sociales, con juegos como el *Buzz!: el Gran Reto* de *PlayStation 2* (Figura 6), que utilizaba unos mandos llamados *Buzzers*, que servían para contestar diversas preguntas sobre temas variados. Debido a la naturaleza desenfadada del videojuego y a su similitud a famosos concursos televisivos de esa época, el juego tuvo gran éxito como título del género *party* y se estrenaron muchas entregas diferentes de la misma saga. Este juego pertenece al subgénero de *quiz games* o juegos de preguntas y respuestas. Otros juegos destacables de este género son *Trivial* o *Jeopardy*.



Figura 6. Captura del videojuego *Buzz!: el Gran Reto*

En los últimos años, el mercado de juegos para móviles ha crecido mucho, y con él, la existencia de juegos *party* y de minijuegos en esta plataforma. Algunos de los títulos más destacados son

*Dumb Ways to Die* o *Among Us*, juegos que se analizan en mayor profundidad en el estudio de mercado. Además, el género *party* ha visto grandes entregas como por ejemplo los juegos *Jackbox Party Pack* u *Overcooked*. Asimismo, con la llegada de la *Nintendo Switch* ha ocurrido algo similar a lo que ocurrió con la *Nintendo Wii*, publicando una gran cantidad de juegos como por ejemplo *51 Worldwide Games* o *Super Mario Party*.

Hoy en día, los videojuegos se han establecido como una forma de socializar y pasar un buen rato, y el género *party* es el mayor exponente de este tipo de juegos sociales. Históricamente, desde el nacimiento del género hace más de 40 años con el título *Olympic Decathlon* hasta ahora, es innegable que la categoría *party* ha prosperado de una forma visible, como se puede observar en la cronología de la Figura 7.



Figura 7. Cronología juegos *party* o de minijuegos

## 2.2 Estudio de mercado

Los juegos de estilo *party*, aunque no son los más populares ni los más jugados entre el público general, sí que son conocidos por la mayoría de jugadores, que prefieren echar unas partidas a estos juegos cuando están con sus amigos. Como ejemplo, solo se tiene que preguntar a cualquier persona a la que le gusten los videojuegos si ha jugado alguna vez al *Mario Party*, o si conoce el *Among Us*.

Desde su lanzamiento en junio de 2018, el juego *Among Us* ha obtenido unos ingresos de 50 millones de dólares [2]. Además, tuvo una media de 60 millones de jugadores activos diarios en octubre de 2020 y 500 millones de jugadores activos mensuales en noviembre de 2020. Estas cifras ayudan a entender el potencial que pueden llegar a tener los juegos de este estilo.

En este apartado se muestra una tabla comparativa (Figura 8) en la que se investigan y estudian los videojuegos que, por su género, plataforma o público objetivo, se asemejan al proyecto de este trabajo y por tanto se consideran sus competidores. A continuación, se hace un análisis de cada videojuego por separado y se evalúan sus puntos fuertes y sus debilidades.

Comparativa Características de productos competidores				
Nombre	Super Mario Party	WarioWare Gold	Among Us	Dumb Ways to Die
Desarrollador	NDcube	Nintendo, Intelligent Systems	Innersloth	Metro Trains Melbourne
Plataformas	Nintendo Switch	Nintendo 3ds	IOS, Android, PC, Nintendo Switch	IOS, Android
Año de lanzamiento	2018	2018	2018	2012
Precio	59,95€	29,95€	3,99€ (PC), Gratis (IOS y Android), 4,29€ (Switch)	Gratis
Género	Party / Minigame	Party / Minigame	Party / Survival	Minigame
Modo un solo jugador	Si	Si	No	Si
Modo Multijugador LAN	Si	No	Si	No
Modo Multijugador online	Si	No	Si	No
Número de jugadores máximo en partida	4	-	10	-
Inicio de sesión con cuentas de terceros	No	No	No	Si
Personajes personalizables	No	No	Si	No
Controles sencillos	Si	Si	No	Si
Tiempo medio por partida (minutos)	60	5	5	1

Figura 8. Tabla comparativa de productos competidores

### 2.2.1 Super Mario Party

La saga *Mario Party*, cuya última entrega se observa en la Figura 9, es el mayor exponente de los juegos del género *party* o de minijuegos. En esta saga, los personajes más famosos de la popular serie de videojuegos *Mario Bros* se unen para competir en una selección muy variada de minijuegos.



Figura 9. Portada de *Super Mario Party*

En *Mario Party* no solo se compete en minijuegos, sino que el modo principal consiste en un tablero por el que se avanza tirando dados como si de un juego de mesa clásico se tratase. El objetivo, como se ve en la Figura 10, es conseguir más monedas y estrellas que los otros jugadores. Estos recursos se consiguen a base de caer en ciertas casillas y ganar los minijuegos correspondientes a esas casillas.



Figura 10. Modo tablero de *Super Mario Party*

La última entrega de esta saga: *Super Mario Party* fue desarrollada por *NDCube* y estrenada en 2018 para *Nintendo Switch* y consideramos que es un competidor directo a *Blitz Party* por atraer al mismo perfil de jugador casual y tener el mismo género de minijuegos. El juego ha vendido 14,79 millones de copias [3]. Con esta cifra y teniendo en cuenta que *Nintendo* no acostumbra a hacer rebajas con sus juegos estimamos que ha recaudado aproximadamente 450 millones de euros.

Lo que convierte a *Super Mario Party* en el principal dominador del género de los minijuegos es principalmente el uso de la propiedad intelectual (IP) de *Mario Bros* y sus populares personajes conocidos y apreciados por la gran mayoría de los jugadores. Además, dado que el juego ha sido creado por un estudio grande, posee una gran calidad tanto gráfica como artística (como apreciamos en la Figura 11). Su última gran ventaja competitiva es la gran cantidad de minijuegos que contiene, cosa que permite a los jugadores invertir mucho tiempo en este juego sin que resulte repetitivo.



Figura 11. Minijuego de *Super Mario Party*

Por otro lado, algunas debilidades que se pueden aprovechar para conseguir una ventaja competitiva son la alta duración de las partidas (alrededor de 60 minutos), cosa que juega en contra del enfoque casual que tienen este tipo de juegos.

Otro punto negativo de *Super Mario Party* es su elevado precio. *Super Mario Party* tiene actualmente un precio de mercado de 59,95€. El público objetivo de este tipo de videojuegos

consiste en jugadores que buscan jugar partidas rápidas y de bajo compromiso, por lo que el elevado despliegue económico que conlleva la compra de este juego es un gran inconveniente.

### 2.2.2 Wario Ware

*Wario Ware*, saga también perteneciente a *Nintendo* y protagonizado por *Wario*, personaje original del universo de *Mario Bros*, es una colección de juegos donde la temática principal consta de una serie de microjuegos a jugar en una rápida sucesión, siendo estos de muy corta duración.



Figura 12. Portada de *Wario Ware Gold*

*Wario Ware Gold* (Figura 12) es la última entrega de la saga, lanzada para la plataforma *Nintendo 3DS* en 2018 y desarrollada por *Nintendo EPD* e *Intelligent Systems*. Respecto a sus anteriores entregas, esta ha añadido nuevos microjuegos. El modo historia de este juego está dividido en cuatro ligas temáticas, cada una de ellas con sus correspondientes microjuegos. Este título vendió aproximadamente 280 mil copias [4] y teniendo en cuenta el precio podemos aproximar que ha recaudado unos 8,39 millones de euros.

Además del modo historia, el juego contiene un modo desafío, en el que se juega una serie con una selección de todos los microjuegos disponibles mezclados de forma aleatoria. El modo desafío contiene opciones de dificultad para los jugadores que busquen un desafío mayor.



Figura 13. Microjuego de *Wario Ware Gold*

Las fortalezas competitivas de *Wario Ware Gold* son la corta duración de sus microjuegos, así como el estilo desenfadado que tiene el juego. Ambas características se aprecian en la Figura 13. Debido a que se puede jugar una partida de manera rápida y que además el juego está disponible en una plataforma portátil como es la *Nintendo 3DS*, los jugadores pueden jugarlo en cualquier momento. Además, el estilo gráfico de los microjuegos es variable y se adapta a cada uno de ellos. Finalmente, el modo historia (Figura 14) tiene una naturaleza de comedia, lo que alimenta ese estilo desenfadado.



Figura 14. Cinemática del modo historia de *Wario Ware Gold*

Como debilidades competitivas destacan la falta de multijugador tanto local como online. Esto puede provocar que los microjuegos, pese a ser variados, se acaben volviendo repetitivos dado

que al no haber un contrincante todas las partidas son iguales. La segunda debilidad competitiva que encontramos en *Wario Ware Gold* es, al igual que en el caso de *Super Mario Party*, su elevado precio. Pese a que es más barato (29,95€), consideramos que sigue siendo un precio demasiado alto para un juego de este estilo.

### 2.2.3 Among Us

*Among Us*, cuya portada se ve en la Figura 15, es un juego multiplataforma online desarrollado por *Innersloth* en 2018. La premisa es la siguiente: en una nave hay un grupo de máximo 10 personas, entre las que se ocultan hasta tres impostores que intentan asesinar al resto de tripulantes.



Figura 15. Portada de *Among Us*

Al inicio de la partida cada jugador recibe un rol que indica lo que tendrá que hacer durante la misma. Si el jugador es un tripulante, habrá de completar pequeñas tareas que funcionan como microjuegos. Si todos los tripulantes completan todas sus tareas habrán ganado la partida. Por otro lado, si el jugador es un impostor, tendrá que asesinar a todos los tripulantes de la nave sin ser descubierto (Figura 16). Si todos los tripulantes son asesinados, los impostores ganan la partida.

El principal motivo del triunfo de este juego es su componente social. Si un jugador encuentra un cadáver en algún punto de la nave puede reportarlo y eso producirá una reunión en la que todos

los jugadores tienen que discutir e intentar descubrir a los impostores o, en el caso de estos impostores, defenderse y convencer al resto de jugadores de que son inocentes.



Figura 16. Impostor asesinando a un tripulante en *Among Us*

A pesar de que *Among Us* se estrenó en 2018, no fue hasta 2020 cuando se volvió realmente popular. Esto se debió probablemente a la cuarentena provocada por la COVID-19, en la que las personas buscaban la interacción social que les faltaba al no poder salir de casa. *Among Us* llegó a las 350 millones de descargas en Noviembre de 2020 [2].

Las fortalezas competitivas de *Among Us* son principalmente su factor social y el hecho de ser gratuito en dispositivos móviles. En el caso del factor social, muchos jugadores de este juego lo hacen mientras están en llamada de voz con sus amigos, y la propia naturaleza del juego de discutir qué jugador es el impostor ayuda a esto. Además, que no sea de pago atrae a muchos jugadores a descargarlo en móvil ya que es fácil acceder a él.



Figura 17. Jugador realizando una tarea en *Among Us*

Por otro lado, su principal debilidad es lo repetitivo que resulta. Las tareas que tienen que completar los tripulantes son pocas y pese a haber dicho anteriormente que eran una suerte de microjuegos, muchas veces resultan más tediosas que divertidas. Por ejemplo, la que se observa en la Figura 17 consiste únicamente en esperar un tiempo fijado. Por esto, hay poca variación entre las diferentes partidas y consideramos este como el principal motivo de que el juego haya tenido un fuerte declive en un periodo de tan solo un año.

#### 2.2.4 Dumb Ways to Die

*Dumb Ways to Die* es un juego para dispositivos móviles estrenado en 2012 y desarrollado por *Metro Trains Melbourne*, responsables de la red de transporte ferroviario de Melbourne. Se originó como una campaña publicitaria con el fin de promover la seguridad cerca de las vías de tren, pero tuvo tal éxito que llevó al desarrollo de diversas secuelas y spin-offs como *Dumb Ways to Die 2: The Games* o *Dumb Ways to Die 3: World Tour*. El juego ha conseguido 130 millones de descargas y la campaña publicitaria de la que forma parte ha conseguido un beneficio de unos 6 millones de dólares en impresiones de los medios [5].

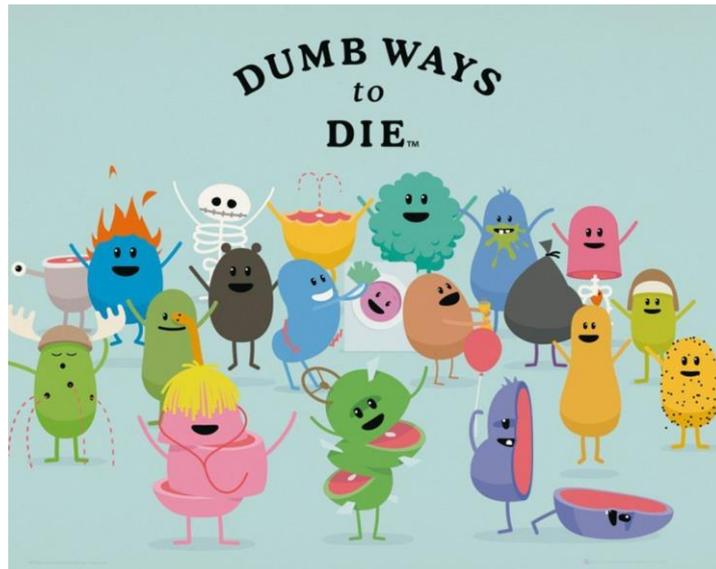


Figura 18. Portada de *Dumb Ways to Die*

*Dumb Ways to Die* goza de un estilo basado en microjuegos frenéticos. El jugador deberá evitar la muerte de los personajes que se observan en la Figura 18. En el caso de perder un microjuego el jugador pierde una vida y al repetirse esto tres veces se acaba la partida. Cuanto más tiempo se mantenga vivo el jugador, más velocidad tendrán los microjuegos, aumentando así su dificultad, pero otorgando más puntos al jugador si consigue superarlos.

Las principales fortalezas de este título son el hecho de que cada microjuego es divertido por sí solo, lo que permite coger el juego en cualquier momento y que sea ameno jugar, aunque sea solo una partida. Además, otro de sus puntos fuertes es su apartado visual y sonoro. Los protagonistas del juego son unos frijoles que, pese a estar sufriendo tragedias, no dejan de cantar y estar felices, lo que refleja la naturaleza poco seria del título e influye en que los jugadores lo perciban como un pasatiempo casual al que se puede jugar en cualquier momento.



Figura 19. Fotograma de la canción *Dumb Ways to Die*

Así mismo, el estreno del juego se acompañó de una canción con el mismo nombre protagonizada por estos personajes (Figura 19). Esta canción tiene actualmente más de 200 millones de reproducciones en *Youtube*, lo que muestra el triunfo del juego y el afecto que los jugadores desarrollaron hacia sus personajes.

Por otro lado, como debilidad destaca principalmente la repetitividad. A pesar de ser entretenidos, el juego posee solamente alrededor de diez microjuegos. Este sería un número aceptable en el caso de que tuvieran profundidad, pero cada uno de ellos tiene una duración de aproximadamente cinco segundos y se basa únicamente en hacer un solo movimiento, lo que hace que se vuelvan aburridos muy rápido. Por ejemplo, el microjuego de la Figura 20 consiste simplemente en tocar la pantalla rápidamente.

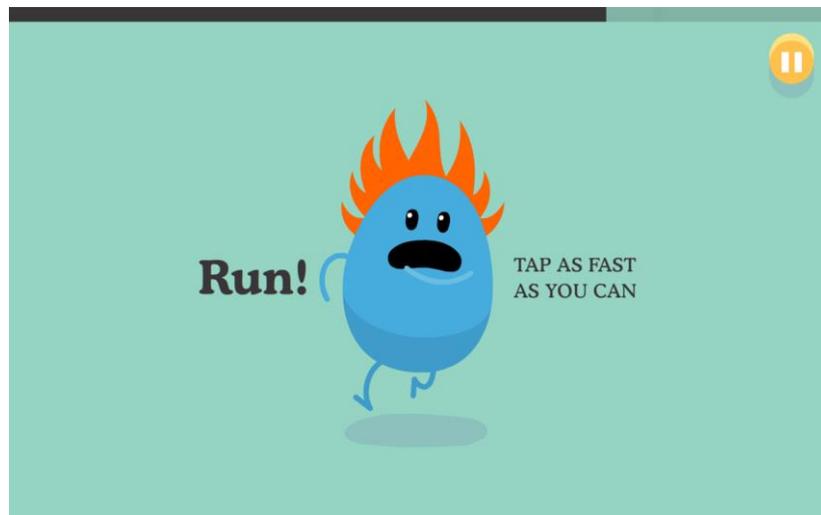


Figura 20. Microjuego de *Dumb Ways to Die*

### 2.3 Análisis DAFO

El análisis DAFO (Figura 21) es una técnica utilizada para evaluar la posición competitiva del producto frente al resto del mercado y a partir de ahí crear una estrategia acorde con los resultados obtenidos. De esta forma, el equipo responsable del producto puede optimizar la manera de afrontar diversos problemas, prepararse frente a las amenazas y centrarse en los puntos fuertes del proyecto.

<p style="text-align: center;"><b>Debilidades</b></p> <ul style="list-style-type: none"> <li>• Experiencia limitada en el desarrollo de videojuegos para dispositivos móviles.</li> <li>• Escasos fondos para el proyecto.</li> <li>• Falta de personal especializado en el apartado visual y sonoro.</li> </ul>	<p style="text-align: center;"><b>Amenazas</b></p> <ul style="list-style-type: none"> <li>• Mercado con mucha competencia.</li> <li>• Funcionalidad limitada del servicio online gratuito.</li> </ul>
<p style="text-align: center;"><b>Fortalezas</b></p> <ul style="list-style-type: none"> <li>• Equipo apasionado por los videojuegos.</li> <li>• Equipo con alta disponibilidad.</li> <li>• Experiencia trabajando juntos como equipo.</li> <li>• Experiencia en el ciclo de desarrollo del software.</li> </ul>	<p style="text-align: center;"><b>Oportunidades</b></p> <ul style="list-style-type: none"> <li>• Mercado con mucho público.</li> <li>• Auge de los juegos competitivos multijugador.</li> <li>• Los juegos basados en minijuegos que existen para dispositivos móviles suelen incluir únicamente multijugador en red local.</li> </ul>

Figura 21. Matriz DAFO de *Blitz Party*

Se puede observar que uno de los puntos clave del análisis es el estado del mercado móvil. Como ya hemos señalado en el epígrafe 1.1, más de la mitad de los beneficios del sector de los videojuegos viene de este mercado y esto ha atraído a múltiples empresas, aumentando así la competencia. Sin embargo, *Blitz Party* goza de ser uno de los pocos de su género con multijugador en línea y esto nos puede dar la oportunidad de penetrar en este mercado.

Por otro lado, aunque el equipo no sea multidisciplinar, la experiencia previa trabajando juntos y la motivación para sacar un buen producto adelante es un factor positivo a tener en cuenta. A partir de esta base se puede crear un proyecto de una calidad considerable aún con falta de financiación.

## 2.4 Modelo de negocio y proyección económica

En lo referido al modelo de negocio, se ha optado por ofrecer el juego sin ningún tipo de coste monetario, siguiendo la concepción de que un juego gratuito atrae a más gente que otro de pago. No obstante, el juego cuenta con un sistema de micropagos con el que se adquiere una cantidad de monedas de juego, bellotas en este caso, a cambio de un precio asequible.

Se prevé la compra de 500 bellotas por 4,99€, o 1100 bellotas por 9,99€. Con estas monedas se obtiene acceso a un repertorio de objetos visuales que se pueden comprar en la tienda interna del juego y equipar posteriormente a nuestro personaje. Aunque no es necesario pagar con dinero real para obtener estos accesorios, sí que acelera mucho el tiempo que tendríamos que dedicar a recolectar bellotas, por lo que es un incentivo importante para la gente a la que le entusiasme vestir a su personaje con prendas nuevas.

Por otra parte, los productos comprados en la tienda del juego son meramente estéticos, por lo que pagar más no dará a ningún jugador una ventaja real sobre su oponente. Como segunda forma de financiación, se prevé la introducción de anuncios en el juego, que se mostrarán en momentos específicos, como al inicio de la aplicación o al final de los minijuegos, y podrán ser eliminados si se paga la mejora a jugador premium, con un coste de 5,99€.

Se cree que el componente *Free-to-Play* del juego atraerá a muchos más jugadores que si fuese de pago, y posteriormente se obtendrán beneficios a partir de los micropagos y los anuncios presentes en la aplicación.

A continuación, se presenta una proyección económica del proyecto a 5 años vista. En el primer trimestre del año de salida del juego, se prevé un gasto inicial en marketing para darle un empujón de salida, y un mayor número de descargas por la publicidad realizada y la página de destacados de *Google Play*. Como se puede observar en la Figura 22, el balance final en los primeros meses resulta negativo, debido a todos estos gastos iniciales.



	Número de Trimestre		
	1	2	3
<b>Datos sobre descargas</b>			
Número de nuevas descargas	Año 1 / T1	Año 1 / T2	Año 1 / T3
Número de nuevas descargas	20000	4000	4000
Dispositivos en los que estaba instalado el juego previamente	0	15000	13000
<b>Total de dispositivos que tienen el juego</b>	20000	19000	17000
<b>Ingresos Trimestrales</b>			
Media anuncios vistos * ganancia por anuncio	1,500 €	1,425 €	1,275 €
Media micropagos hechos * ganancia media por micropago	1,000€	950€	850€
<b>Total ingresos</b>	2,500 €	2,375 €	2,125 €
<b>Gastos Trimestrales</b>			
Infraestructura Cloud	360 €	360 €	360 €
Licencias Software	0 €	0 €	0 €
Ordenadores	0 €	0 €	0 €
Muebles oficina e instalaciones	0 €	0 €	0 €
Marketing	10,000 €	5,000 €	5,000 €
Alquiler oficina	0 €	0 €	0 €
Internet, electricidad, agua, teléfono, etc.	300 €	300 €	300 €
Gestoría	0 €	0 €	0 €
Director Ejecutivo	6,000 €	6,000 €	6,000 €
Director Técnico	6,000 €	6,000 €	6,000 €
Director de Arte	6,000 €	6,000 €	6,000 €
Director Narrativo	6,000 €	6,000 €	6,000 €
Desarrolladores Senior	6,000 €	6,000 €	6,000 €
Técnicos de soporte	0 €	0 €	0 €
Administrativo	0 €	0 €	0 €
<b>Total Gastos</b>	40,660 €	35,660 €	35,660 €
<b>Resultado Trimestral</b>			
	-38,160 €	-33,285 €	-33,535 €
<b>Resultado Trimestral Acumulado</b>			
	-38,160 €	-71,445 €	-104,980 €

Figura 22. Proyección económica año 1

A lo largo del primer año se espera una decaída en el número de descargas y jugadores, por lo que el primer trimestre del segundo año se procede a aumentar el presupuesto destinado al marketing, con la finalidad de reflotar el juego. A partir de este punto, se espera que aumente exponencialmente el número de jugadores, como aparece en la *Figura 23*. Con estos resultados, se planifica conseguir un espacio físico y aumentar la plantilla del equipo (*Figura 24*).

4	5	6	7	8	9	10
Año 1 / T4	Año 2 / T1	Año 2 / T2	Año 2 / T3	Año 2 / T4	Año 3 / T1	Año 3 / T2
4000	15000	50000	40000	32000	25000	20000
12000	10000	20000	45000	70000	90000	100000
16000	25000	52000	85000	102000	115000	120000
1,200 €	1,875 €	39,000 €	63,750 €	76,500 €	86,250 €	90,000 €
800€	1,250€	26,000€	42,500€	51,000€	57,500€	60,000€
2,000 €	3,125 €	65,000 €	106,250 €	127,500 €	143,750 €	150,000 €
360 €	360 €	1,000 €	1,000 €	1,360 €	1,360 €	1,360 €
0 €	0 €	3,000 €	0 €	0 €	3,000 €	3,500 €
0 €	0 €	2,000 €	2,000 €	0 €	0 €	1,000 €
0 €	0 €	800 €	0 €	0 €	0 €	200 €
5,000 €	10,000 €	10,000 €	15,000 €	15,000 €	15,000 €	15,000 €
0 €	0 €	4,500 €	4,500 €	4,500 €	4,500 €	4,500 €
300 €	300 €	600 €	600 €	600 €	600 €	600 €
0 €	0 €	500 €	900 €	900 €	900 €	900 €
6,000 €	6,000 €	6,000 €	6,000 €	6,000 €	10,000 €	10,000 €
6,000 €	6,000 €	6,000 €	6,000 €	6,000 €	10,000 €	10,000 €
6,000 €	6,000 €	6,000 €	6,000 €	6,000 €	10,000 €	10,000 €
6,000 €	6,000 €	6,000 €	6,000 €	6,000 €	10,000 €	10,000 €
6,000 €	6,000 €	6,000 €	18,000 €	18,000 €	21,000 €	21,000 €
0 €	0 €	3,000 €	3,000 €	3,000 €	3,000 €	3,000 €
0 €	0 €	3,000 €	3,000 €	3,000 €	3,000 €	3,000 €
35,660 €	40,660 €	58,400 €	72,000 €	70,360 €	92,360 €	94,060 €
-33,660 €	-37,535 €	6,600 €	34,250 €	57,140 €	51,390 €	55,940 €
-138,640 €	-176,175 €	-169,575 €	-135,325 €	-78,185 €	-26,795 €	29,145 €

Figura 23. Proyección económica año 2

Personal contratado	Año 1 / T1	Año 1 / T2	Año 1 / T3	Año 1 / T4	Año 2 / T1	Año 2 / T2	Año 2 / T3	Año 2 / T4	Año 3 / T1	Año 3 / T2
Director Ejecutivo	1	1	1	1	1	1	1	1	1	1
Director Técnico	1	1	1	1	1	1	1	1	1	1
Director de Arte	1	1	1	1	1	1	1	1	1	1
Director Narrativo	1	1	1	1	1	1	1	1	1	1
Desarrolladores Senior	1	1	1	1	1	1	3	3	3	3
Técnicos de soporte	0	0	0	0	0	1	1	1	1	1
Administrativo	0	0	0	0	0	1	1	1	1	1
<b>Total empleados</b>	<b>5</b>	<b>5</b>	<b>5</b>	<b>5</b>	<b>5</b>	<b>7</b>	<b>9</b>	<b>9</b>	<b>9</b>	<b>9</b>

Figura 24. Personal contratado



Durante los años 3 y 4 se proyecta que el juego alcance su punto más alto, llegando a tener 600.000 descargas nuevas en el segundo trimestre del cuarto año y un total de 1.710.000 descargas el cuarto trimestre del cuarto año, como se muestra en la Figura 25. Con los ingresos obtenidos se procede a la contratación de más personal y al aumento de sueldos de los empleados. Adicionalmente, se llevan a cabo campañas de *marketing* y se introduce contenido adicional al juego para fomentar tanto la llegada de jugadores nuevos como el retorno de los más veteranos.

11	12	13	14	15	16	17
Año 3 / T3	Año 3 / T4	Año 4 / T1	Año 4 / T2	Año 4 / T3	Año 4 / T4	Año 5 / T1
180000	150000	100000	600000	490000	360000	250000
1010000	800000	700000	750000	1200000	1350000	1400000
1190000	950000	800000	1350000	1690000	1710000	1650000
89,250 €	71,250 €	60,000 €	101,250 €	126,750 €	128,250 €	123,750 €
59,500€	47,500€	40,000€	67,500€	84,500€	85,500€	82,500€
148,750 €	118,750 €	100,000 €	168,750 €	211,250 €	213,750 €	206,250 €
1,360 €	1,360 €	1,360 €	1,360 €	1,360 €	1,360 €	1,360 €
0 €	0 €	4,500 €	6,000 €	0 €	0 €	4,500 €
0 €	0 €	3,000 €	0 €	0 €	0 €	2,000 €
0 €	0 €	300 €	0 €	0 €	0 €	800 €
15,000 €	15,000 €	15,000 €	25,000 €	15,000 €	15,000 €	15,000 €
4,500 €	4,500 €	6,000 €	6,000 €	6,000 €	6,000 €	6,000 €
600 €	600 €	1,200 €	1,200 €	1,200 €	1,200 €	1,200 €
900 €	900 €	1,200 €	1,200 €	1,200 €	1,200 €	1,200 €
10,000 €	10,000 €	10,000 €	10,000 €	10,000 €	10,000 €	10,000 €
10,000 €	10,000 €	10,000 €	10,000 €	10,000 €	10,000 €	10,000 €
10,000 €	10,000 €	10,000 €	10,000 €	10,000 €	10,000 €	10,000 €
10,000 €	10,000 €	10,000 €	10,000 €	10,000 €	10,000 €	10,000 €
21,000 €	21,000 €	34,000 €	34,000 €	34,000 €	46,000 €	46,000 €
3,000 €	3,000 €	6,000 €	6,000 €	6,000 €	9,000 €	9,000 €
3,000 €	3,000 €	3,000 €	3,000 €	3,000 €	4,500 €	4,500 €
89,360 €	89,360 €	115,560 €	123,760 €	107,760 €	124,260 €	131,560 €
59,390 €	29,390 €	-15,560 €	44,990 €	103,490 €	89,490 €	74,690 €
88,535 €	117,925 €	102,365 €	147,355 €	250,845 €	340,335 €	415,025 €

Figura 25. Proyección económica años 3 y 4

Finalmente, el quinto año supone una disminución progresiva del número de jugadores, pasando a tener el último trimestre de este año tan solo 980.000 dispositivos con el juego instalado, frente a los 1.710.000 del año anterior (Figura 26).

Datos sobre descargas	Año 5 / T2	Año 5 / T3	Año 5 / T4
Número de nuevas descargas	180000	120000	60000
Dispositivos en los que estaba instalado el juego previamente	1300000	1100000	920000
<b>Total de dispositivos que tienen el juego</b>	<b>1480000</b>	<b>1220000</b>	<b>980000</b>

Figura 26. Número de descargas año 5

La proyección económica se ha realizado a cinco años vista dado que gracias a campañas publicitarias y una constante evolución del videojuego, se espera que la vida del producto llegue a los cinco años con una base de jugadores sólida. Para ello, se realizarán campañas estacionales en las que se introducirá contenido adicional (como accesorios adicionales en la tienda o nuevos minijuegos temáticos) relacionados con la festividad correspondiente. Como ejemplos, se añadirán eventos para Navidad, Halloween, Pascua, etc.

En cuanto a los ingresos, y como se puede observar en la Figura 27, no se empiezan a obtener beneficios hasta casi el octavo trimestre. Los gastos iniciales se deben, mayoritariamente, a los salarios de los empleados, el *marketing* inicial y otros gastos menores relacionados con la infraestructura de la empresa (gastos de oficina, servidores *cloud*, etc.).



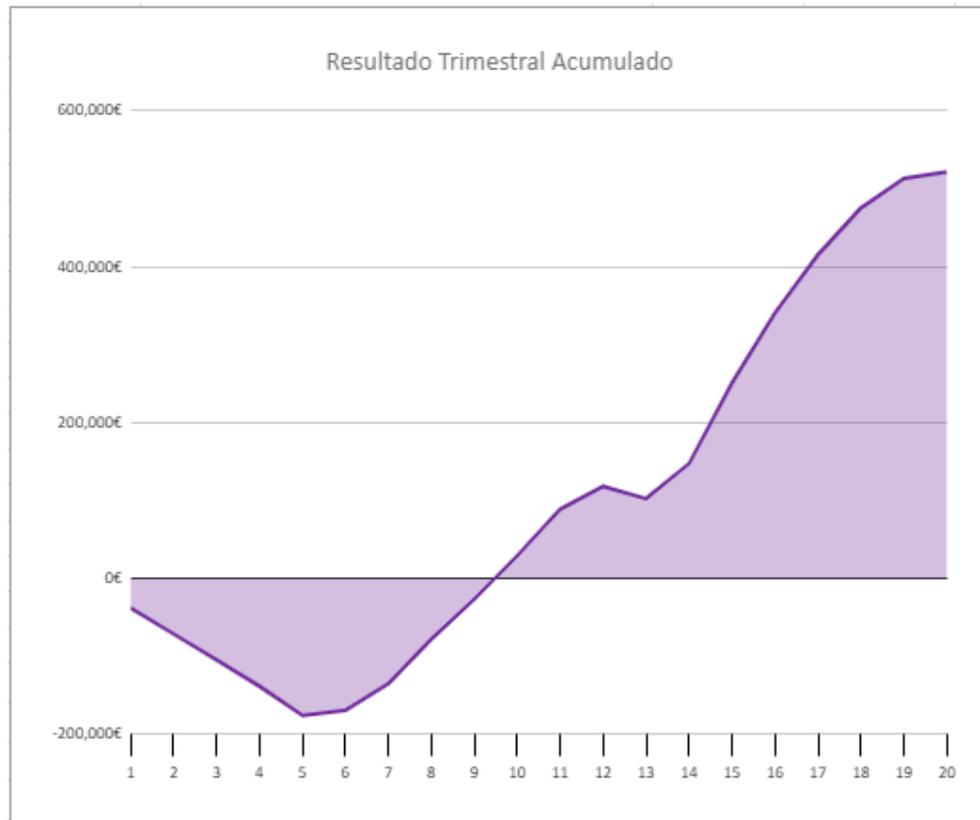


Figura 27. Resultado acumulado de ingresos

## 2.5 Lean Canvas

El *Lean Canvas* (Figura 28) es una herramienta de visualización de modelos de negocio para maximizar el índice de probabilidades de éxito. Consta de 9 secciones en las que se estudian aspectos clave para la buena acogida y la rentabilidad del producto. A partir de estos puntos, se espera focalizar la atención en aquellos apartados importantes que se han detectado, generando una estrategia capaz de generar el máximo beneficio posible.

<p><b>2 Problema</b></p> <p>En el mercado de juegos móviles, tanto los juegos estilo <i>party</i> como los juegos estilo minijuegos son un terreno bastante inexplorado.</p> <p>Algunos posibles competidores son <i>Among Us</i>, <i>Wario Ware</i>, <i>Mario Party</i>, <i>Fall Guys</i> o <i>Clash Royale</i>. Entre estos juegos, algunos son casuales, algunos son multijugador y algunos son estilo <i>party</i>. Pero ninguno de ellos cumple todas estas características.</p>	<p><b>4 Solución</b></p> <p>Diseñar un juego gratuito que sea fácil de aprender, de partidas rápidas y dinámicas, en el que puedas divertirte con otras personas.</p>	<p><b>3 Proposición de valor</b></p> <p>El único juego para móvil en el que podrás desafiar a jugadores de todo el mundo a trepidantes batallas de minijuegos.</p>	<p><b>9 Ventaja competitiva</b></p> <ul style="list-style-type: none"> <li>-Posibilidad de jugar dónde y cuándo sea, gracias a partidas rápidas y al dispositivo móvil.</li> <li>-Apto para jugar con amigos y familia, debido al fácil aprendizaje y al componente multijugador.</li> </ul>	<p><b>1 Clientes</b></p> <ul style="list-style-type: none"> <li>-El público objetivo son jugadores a partir de 7 años con acceso a dispositivos móviles ya sean <i>smartphones</i> o <i>tablets</i>.</li> <li>-Debido a la naturaleza del videojuego, principalmente queremos atraer al mercado de jugadores casuales que no invierten demasiado tiempo en este sector.</li> </ul>
	<p><b>8 Métricas</b></p> <ul style="list-style-type: none"> <li>- Número de descargas en las tiendas digitales.</li> <li>- Porcentaje de dispositivos que mantienen la app instalada.</li> <li>- Porcentaje de jugadores que adquieren elementos del juego.</li> </ul>		<p><b>5 Canales</b></p> <ul style="list-style-type: none"> <li>- Medio de venta a través de <i>Google Play Store</i> y <i>App Store</i>.</li> <li>- Publicidad mediante redes sociales como <i>Twitter</i> o <i>Instagram</i>, <i>Youtube</i> y canales de comunicación tales como servidores de <i>Discord</i>.</li> </ul>	<ul style="list-style-type: none"> <li>-<i>Early adopters</i>: amigos y familiares de los desarrolladores.</li> <li>-El modelo de negocio es <i>Free to Play</i>, es decir, el juego es gratuito pero contiene elementos cosméticos de pago que los jugadores podrán adquirir si lo desean.</li> </ul>



## 7 Costos

Algunos costos fijos son la infraestructura *cloud* y los salarios de los empleados mientras que uno de los costes variables más importantes es el *marketing*:

- Costes medios por trimestre esperados en el primer año: 25000€
- Costes medios por trimestre esperados en el segundo año: 50000€

## 6 Ingresos

La principal fuente de ingresos se sostendrá en base a un sistema de anuncios entre partidas o dentro de la misma, así como micropagos incluidos en la tienda para la adquisición de elementos del juego.

· Primer año:

- Ingresos medios esperados trimestrales por visualizaciones de anuncios: 5400€
- Ingresos medios esperados trimestrales por microtransacciones realizadas: 3600€

· Segundo año:

- Ingresos medios esperados trimestrales por visualizaciones de anuncios: 181.125€
- Ingresos medios esperados trimestrales por microtransacciones realizadas: 120.750€

Figura 28. *Lean Canvas* de *Blitz Party*

## 2.6 Conclusiones

Viendo el análisis de los competidores del mercado observamos que, a pesar de ser un género con varios exponentes, falta la capacidad de jugar online en todos ellos. Debido a esto, el género de los minijuegos online es un nicho que *Blitz Party* puede llenar y así conseguir un número elevado de jugadores.

Además, en el análisis DAFO y el Lean Canvas destaca que el mercado móvil tiene una gran cantidad de potenciales clientes, por tanto, consideramos que es el que más se adapta a los planes de lanzamiento y el que otorga al proyecto la mayor posibilidad de alcanzar el éxito.

Finalmente, tras valorar el modelo de negocio y la proyección económica, se espera que las ventajas competitivas del producto deriven, tras una campaña de *marketing* y la posterior adaptación al mercado, en beneficios económicos.



## 3. Desarrollo de la idea de negocio

---

Este proyecto ha sido desarrollado en un periodo de tiempo de 6 meses, empezando en enero de 2021. No obstante, la concepción de la idea y la búsqueda de tutor tuvieron lugar unos meses antes.

El concepto inicial del videojuego se originó a partir de una tarde en la que el equipo de desarrollo estaba jugando a un juego de estilo *party*. Surgió la idea de que este estilo de videojuegos es dinámico y muy divertido, sobre todo cuando se puede disfrutar con amigos. Se propuso la realización de un videojuego de estas características, que tuviese multijugador online y se pudiese instalar en dispositivos móviles.

Con estos aspectos clave como punto de partida, y la intención de encontrar un tutor para el TFG, contactamos con el profesor de la asignatura Introducción a la Programación de Videojuegos y este, al no poder tutelar a más alumnos, nos puso en contacto con Francisco José Abad, profesor de la asignatura de Entornos de Desarrollo de Videojuegos (EDV). En la reunión que tuvo lugar el 9 de noviembre de 2020, se presentó la idea en forma de Pitch Doc (Figura 29).

A partir de enero se llevó a cabo el estudio de mercado, el *Lean Canvas* y el *Game Design Document* (GDD); y en febrero se empezó el desarrollo del videojuego en paralelo a Entornos de Desarrollo de Videojuegos. En esta asignatura se pedía programar un videojuego en 2D utilizando Unity, y se explicaban una serie de temas que facilitan el desarrollo de este tipo de juegos. Por tanto, se nos propuso presentar este trabajo como proyecto de dicha asignatura, pudiendo ser mostrado ante los demás compañeros y evaluado por el profesor.

Además, nos reunimos con Patricio Letelier, profesor de las asignaturas Proceso de Software (PSW) y Proyecto de Ingeniería de Software (PIN). Nos permitió utilizar la herramienta Worki para organizar el trabajo en equipo utilizando metodologías ágiles y nos dió la oportunidad de formar parte del espacio de emprendimiento de la escuela, Start.inf.

Durante el transcurso de la asignatura, se presentó el progreso realizado en tres fechas distintas y se obtuvo *feedback* por parte del profesor, pudiendo mejorar así la aplicación. La última de las presentaciones, en mayo, supuso el primer *Minimum Viable Product* o MVP del videojuego, que se publicó en la *Play Store* con la finalidad de aumentar el alcance del producto. Los conceptos aprendidos fueron muy útiles y derivaron en unos resultados sobresalientes.

Al terminar la asignatura, el desarrollo continuó a buen ritmo y se realizó un segundo MVP el 28 de junio con una versión con funcionalidad completa. Finalmente, se publicó una actualización en la *Play Store* con dicha versión finalizada.

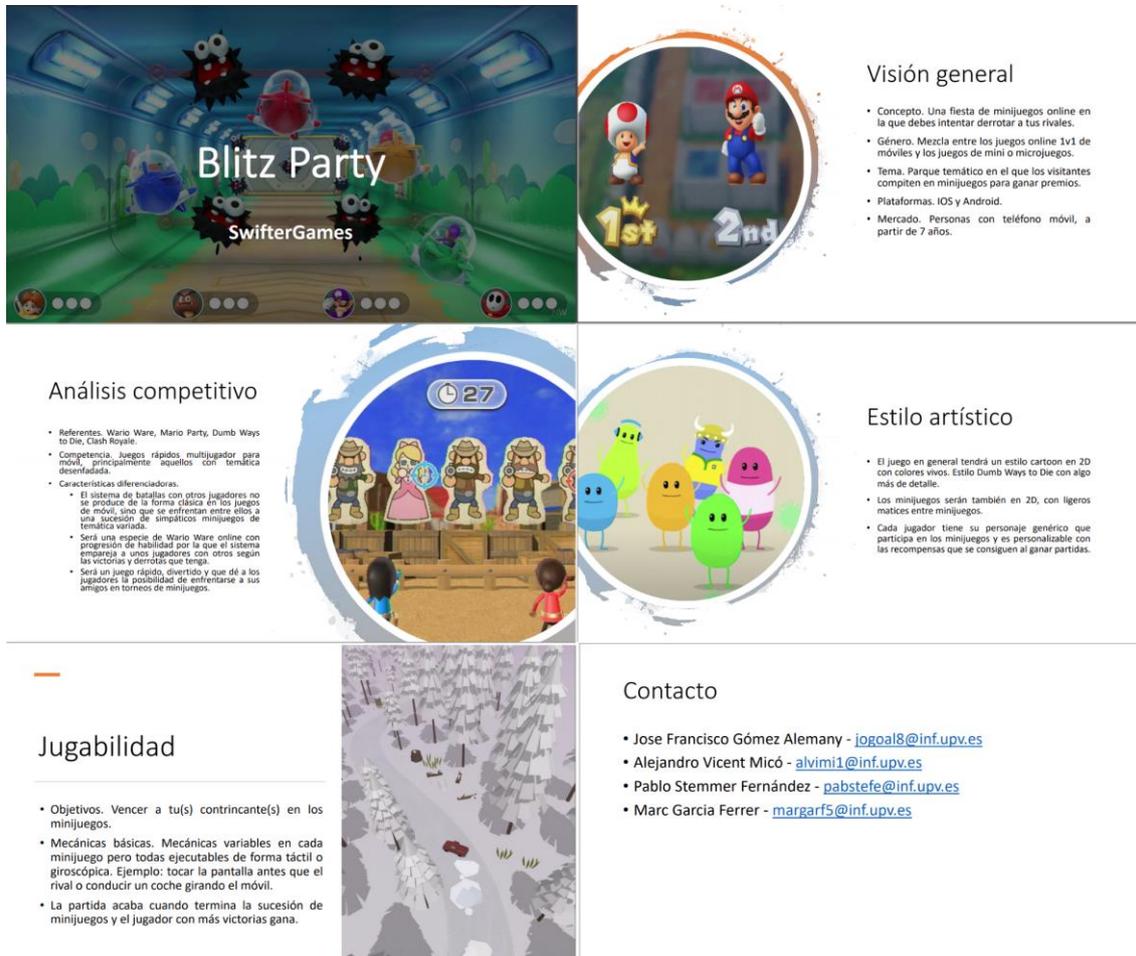


Figura 29. *Pitch Doc*

### 3.1 Mapa de características

Al inicio del proyecto se llevó a cabo una tormenta de ideas, a partir de la cual se ha elaborado un mapa de características que se ha usado a la hora de desarrollar el juego para determinar qué funcionalidades incluir en el producto. Se han hecho dos MVPs con sus consiguientes experimentos y a continuación, en la Figura 30, se observa el mapa de características con todas las funcionalidades que se prevé implementar.



Figura 30. Mapa de características de *Blitz Party*

### 3.2 Primer *Minimum Viable Product*

Para el primer MVP, se ha decidido desarrollar todo lo relacionado con las bases del juego y el modo de un solo jugador. Se puede observar las características seleccionadas en la Figura 31. Como se puede apreciar, en este MVP se han incluido las características sobre la interfaz del juego, la cual está compuesta por diferentes pestañas. Esta interfaz incluye la tienda de accesorios, la pestaña de personalización del personaje y el menú de selección de minijuegos, entre otros.

Como se acaba de mencionar, se buscaba tener el modo de un solo jugador completo y por ello se decidió implementar tres minijuegos: *Binky Pursuit*, *Whack-A-Mole* y *Cowboy Duel* con su jugabilidad únicamente *singleplayer*. Por este motivo, se han incluido las mecánicas de juego, el comportamiento de los enemigos y lo referente a los aspectos sonoros y visuales de los tres minijuegos. Además, se han creado ciertos sistemas usados por todos los minijuegos como por ejemplo el cronómetro, la puntuación y el menú de pausa.

Por otro lado, se han añadido los sistemas de recompensas, obtenidas cuando los jugadores resultan victoriosos en los minijuegos, y el sistema de compra e inventario para que los jugadores puedan darles uso a estas recompensas. Finalmente, se ha agregado el modo batalla, un modo de juego formado por tres minijuegos en orden aleatorio.

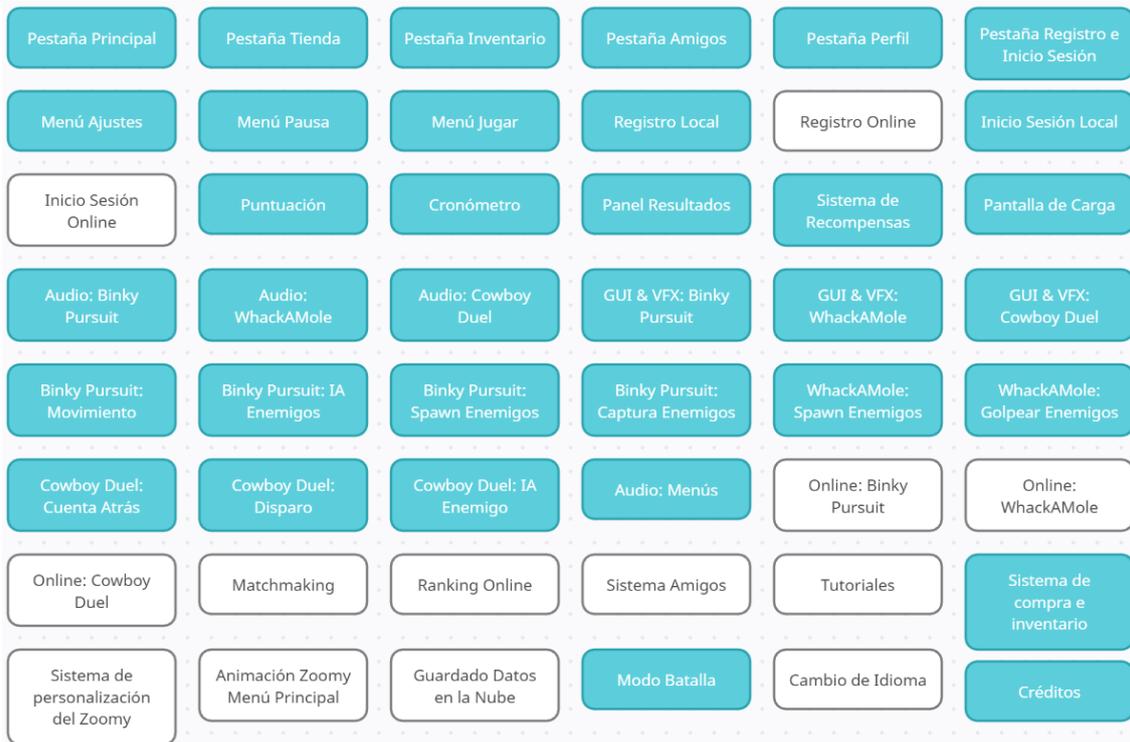


Figura 31. Mapa de características del primer MVP

### 3.2.1 Desarrollo del MVP

Para este primer MVP, se han elegido tres de los minijuegos de la lista que se ha ideado durante la sesión de *brainstorming* inicial (Figura 32). Se ha intentado buscar variedad en la jugabilidad y las mecánicas y, por supuesto, que sea divertido tanto jugando contra una IA como contra otro jugador. Con estos criterios, se ha decidido comenzar el desarrollo con los minijuegos: *Binky Pursuit*, *Whack-A-Mole* y *Cowboy Duel*.



Figura 32. Minijuegos seleccionados del primer MVP.

A continuación, se comentan las características principales de los minijuegos elegidos para este primer MVP:



- *Binky Pursuit* (Figura 33): Un juego basado en capturar más *Binkies* que el rival, una IA. El jugador hace uso de un joystick que aparece con el primer contacto con la pantalla para mover al personaje y de un botón para capturar a los *Binkies*. Estas criaturas aparecen de sus madrigueras cada poco tiempo y se mueven por la superficie. Capturar a cada uno de estos *Binkies* recompensa al jugador con un punto.

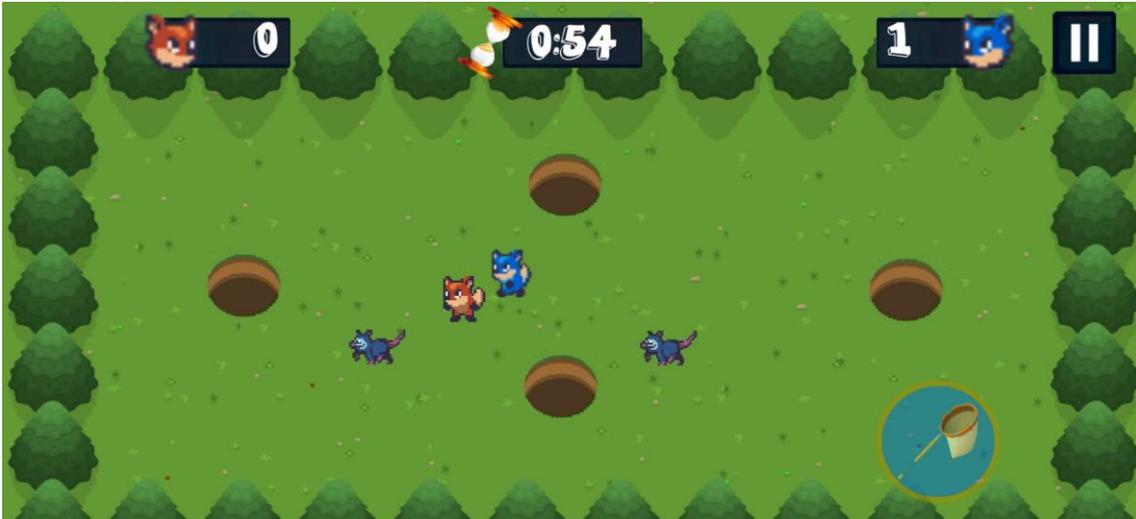


Figura 33. Captura del minijuego *Binky Pursuit*.

- *Whack-A-Mole* (Figura 34): Este minijuego está basado en el clásico juego de recreativa con el mismo nombre. El jugador debe golpear a los topos y a los topos dorados simplemente tocando la pantalla para obtener puntos y así intentar superar la puntuación establecida. Se debe tener cuidado ya que ocasionalmente aparecen *Zoomies* que restan puntuación al jugador si este les golpea.

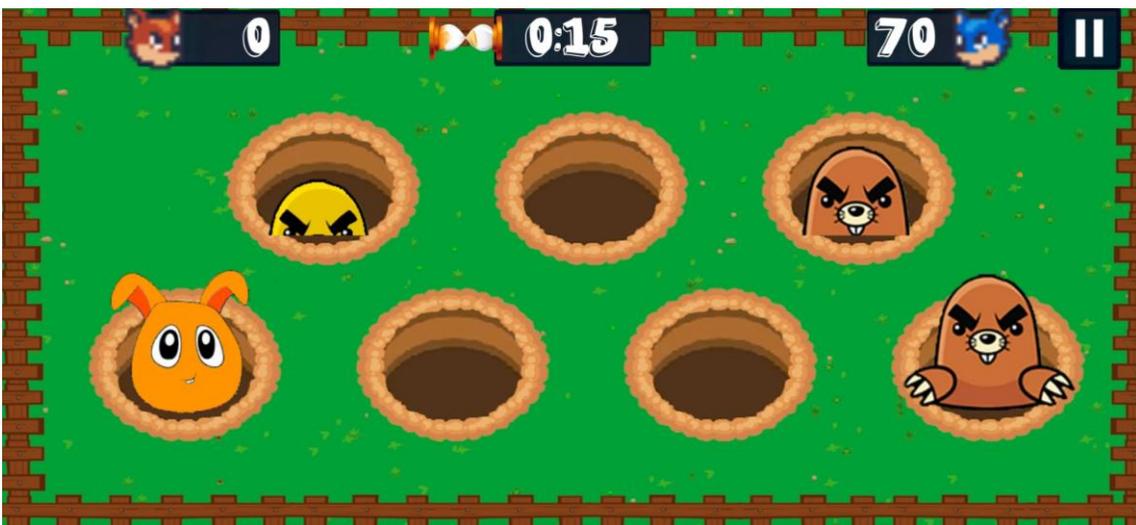


Figura 34. Captura del minijuego *Whack-A-Mole*.

- *Cowboy Duel* (Figura 35): Un juego en el que el jugador debe reaccionar rápidamente para pulsar la pantalla antes que su rival cuando aparezca la señal de disparo. En caso de que el jugador dispare antes de tiempo, fallará el disparo y perderá la ronda automáticamente. El primer jugador que se alcance victorioso en dos rondas obtendrá la victoria.



Figura 35. Captura del minijuego *Cowboy Duel*.

### 3.2.2 Experimento

Para el primer experimento se ha decidido elaborar una encuesta con la mayoría de respuestas cerradas, algunas en una escala del 1 al 10 y otras finales abiertas para conseguir nuevas ideas. La difusión de esta encuesta se ha llevado a cabo a través de las redes sociales personales de los miembros del equipo con la intención de que llegara a familiares, amigos y conocidos. Se recibió bastante difusión con 50 *retweets* y se llegaron a conseguir 39 respuestas para la encuesta.



Figura 36. Tuit de promoción del primer MVP

Con el tuit de la Figura 36 se ha conseguido llegar a un número de respuestas sustancial que ha ayudado a ver las áreas del proyecto que necesitaban cambios y las que les gustaban a los jugadores. A continuación, se muestran las respuestas obtenidas en el experimento y se detalla cómo se han usado para hacer cambios e incluir mejoras.

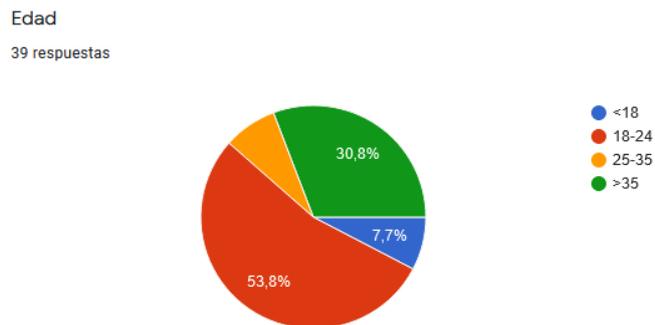


Figura 37. Edad de los participantes de la encuesta

En la Figura 37 se observa que, pese a que hay jugadores de todos los grupos de edad, el principal ha sido el que está comprendido por personas de entre 18 y 24 años, cosa que se explica mayormente porque son las amistades y conocidos de los miembros del equipo. Destaca también el grupo de mayores de 35 años, que en líneas generales han dado respuestas muy favorables a la mayoría de las preguntas.

¿Cuántas horas a la semana dedicas de media a jugar a videojuegos?

39 respuestas

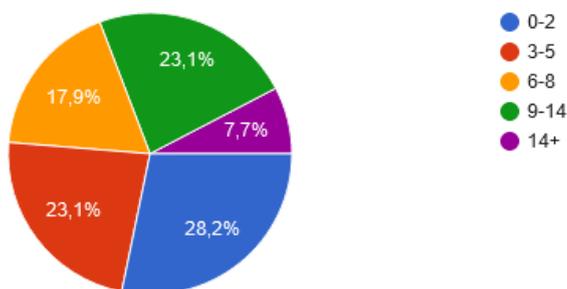


Figura 38. Tiempo que dedican a videojuegos los participantes de la encuesta

¿Sueles jugar a videojuegos en dispositivos móviles?

39 respuestas

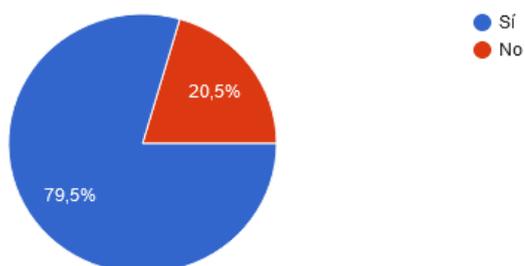


Figura 39. Porcentaje de participantes que juegan en dispositivos móviles

Resalta en la Figura 38 que más del 50% de los participantes dedican menos de 5 horas semanales a jugar a videojuegos de media. De aquí se deduce que *Blitz Party* es un juego apto para todo el mundo, no solo para jugadores habituales. Además, en la Figura 39 se observa que la mayoría de gente juega en dispositivos móviles, lo que denota que es un mercado muy grande y que casi cualquier persona es un posible cliente.

En caso afirmativo, ¿cuántas de las horas semanales mencionadas anteriormente son jugando en móviles?

33 respuestas

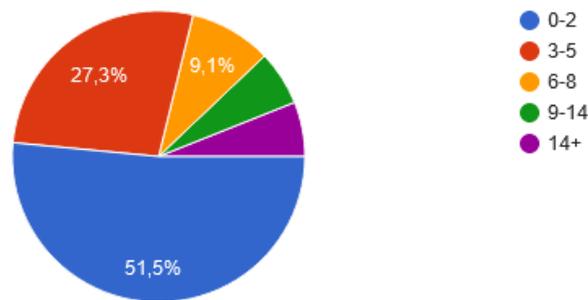


Figura 40. Horas que juegan los participantes de la encuesta en dispositivos móviles

En la Figura 40 vemos que más del 75% de los encuestados juegan en dispositivos móviles menos de 5 horas a la semana. Esto se corresponde con nuestras expectativas del mercado móvil y se ha desarrollado el proyecto teniéndolo en cuenta. *Blitz Party* es un juego de partidas rápidas para el que no se necesita mucho tiempo ni dedicación comparado a juegos de otros géneros como por ejemplo los MMO.

De los 3 minijuegos, ¿cuál ha sido el que más te ha gustado?

39 respuestas

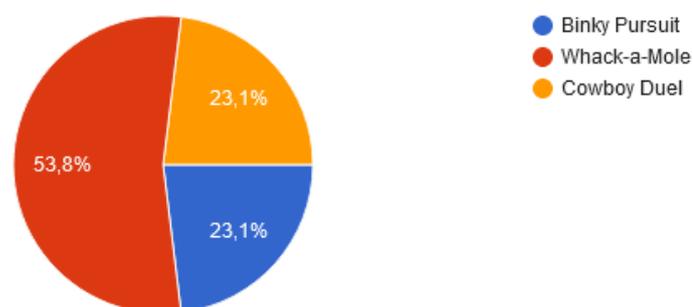


Figura 41. Minijuego favorito de los encuestados

Se puede observar en la Figura 41 que, de los tres minijuegos, el mejor valorado ha sido *Whack-a-Mole*. Se piensa que el motivo de esto es que es el minijuego más sencillo de los tres y esto hace que los jugadores, al haberlo aprendido más rápido, tengan una opinión más positiva sobre él. Otra teoría es que, al no ser una competición contra otro jugador o IA, sino que es una puntuación a batir los jugadores lo consideran más relajante o más divertido.

Del 1 al 10, ¿cómo calificarías la curva de aprendizaje del juego?

39 respuestas

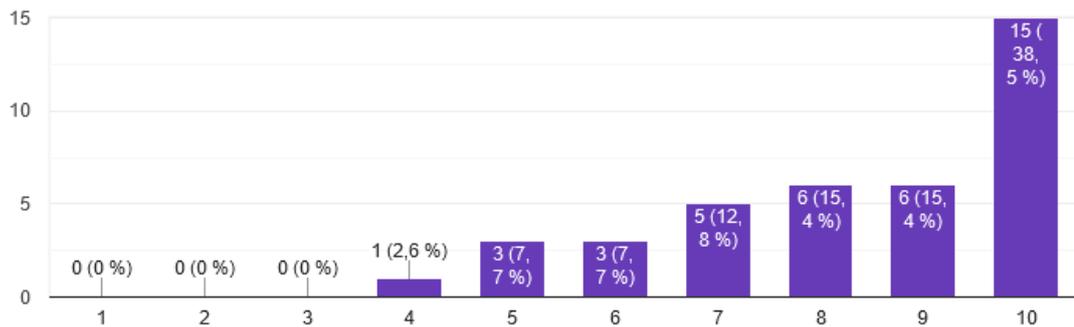


Figura 42. Puntuaciones de la curva de aprendizaje

Del 1 al 10, ¿cómo calificarías la dificultad del juego?

39 respuestas

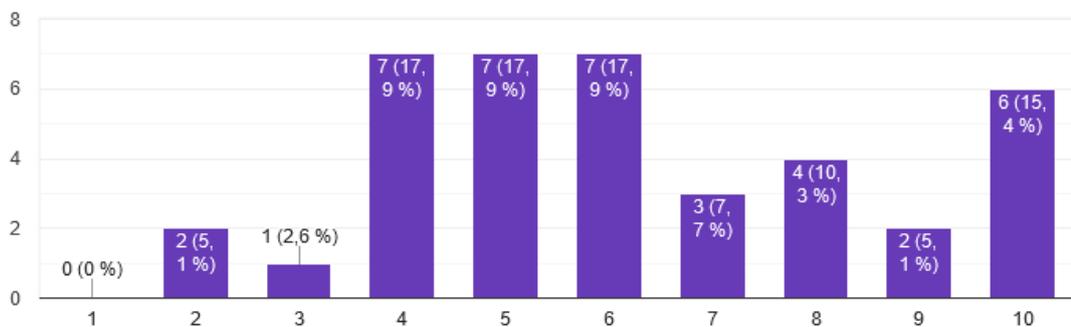


Figura 43. Puntuaciones sobre la dificultad del juego

En la Figura 42, siendo el uno “Muy difícil aprender a jugar” y el diez “Muy fácil aprender a jugar”, la mayoría de encuestados consideran que el juego es fácil de aprender, aunque se observan también algunas respuestas del orden de cuatro, cinco o seis que demuestran que para otros existen dificultades a la hora de interiorizar los controles. Estas respuestas se consideran importantes dado que en el producto final al que se busca llegar es de vital importancia que los controles sean fáciles para todos los jugadores, no solo para la mayoría.

Sin embargo, en la Figura 43, siendo el uno “Demasiado fácil” y el diez “Demasiado difícil” vemos que las respuestas predominan entre el cuatro, cinco y seis, destacando una mayor cantidad de respuestas en la mitad alta. De aquí extraemos que, pese a que generalmente la dificultad sea acertada, se pueden hacer algunos cambios de balance para facilitar un poco los minijuegos.

Sobre la pregunta anterior, ¿alguno de los 3 minijuegos te parece demasiado difícil?

24 respuestas

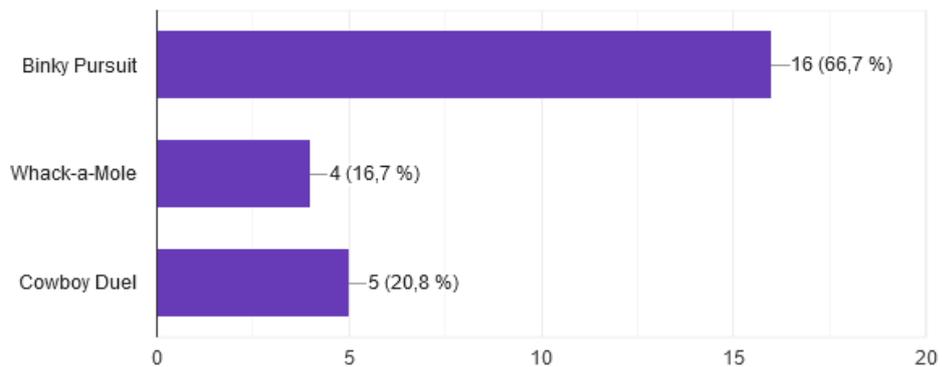


Figura 44. Opinión sobre dificultad excesiva de algún minijuego

Se distingue en la Figura 44 que 16 de los encuestados consideran que la dificultad de Binky Pursuit es demasiado elevada. Debido a esto, se ha decidido reducir la velocidad de la IA para facilitar la partida. Además, debido a que se ha aumentado ligeramente la dificultad de Cowboy Duel añadiendo la posibilidad de fallar si se dispara antes de tiempo, se ha decidido aumentar el tiempo que tarda el rival en disparar para que no sea excesivamente difícil.

En el apartado visual, ¿cómo calificarías del 1 al 10 la claridad de la interfaz de usuario?

39 respuestas

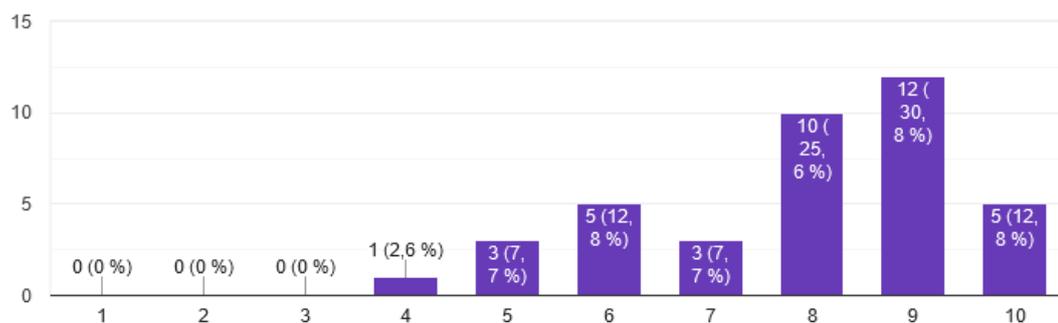


Figura 45. Opiniones sobre la claridad de la interfaz gráfica

En la Figura 45 se observa que, siendo el uno “poco clara” y el diez “muy clara”, mayormente la claridad de la interfaz de usuario se considera buena y observamos que una gran parte de las respuestas son de ocho y nueve. Por eso, hemos decidido mantener la interfaz gráfica en el mismo estado.

¿Crees que el diseño de los personajes encaja con el estilo y la ambientación del juego?

39 respuestas

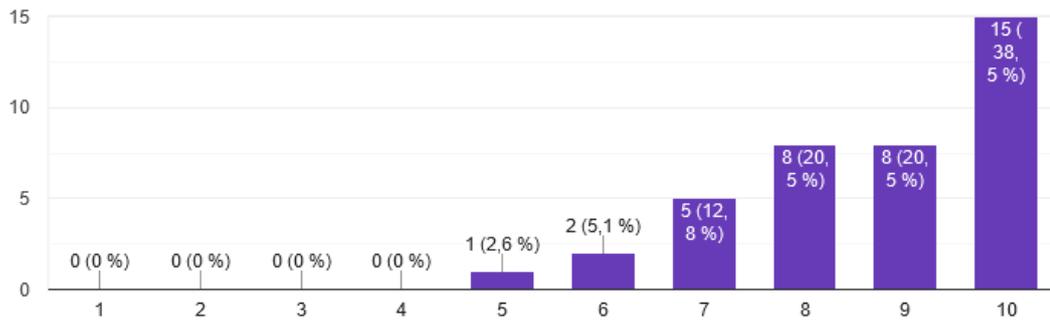


Figura 46. Opiniones sobre diseño y ambientación

La Figura 46 destaca por tener puntuaciones muy altas. Siendo el uno “no encajan” y el diez “encajan mucho”, todas las respuestas son de cinco o más, siendo casi todas de ocho, nueve y diez. Por esto, hemos decidido mantener el diseño tal como está.

En cuanto a la música, ¿te parece acertada la elección de las canciones para cada minijuego?

39 respuestas

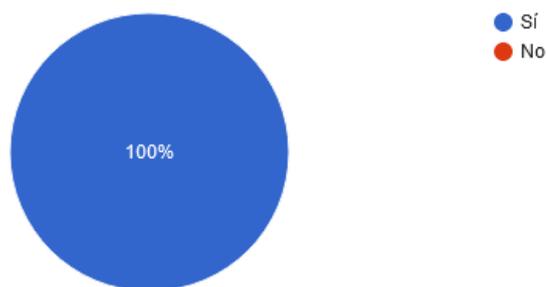


Figura 47. Opinión sobre la música de los minijuegos

En la Figura 47 vemos que al 100% de los encuestados les parece acertada la elección de la música así que se ha decidido no cambiar nada, únicamente añadir efectos de sonido para dar mayor profundidad sonora al juego.



Del 1 al 10, ¿qué puntuación le darías a Blitz Party?

39 respuestas

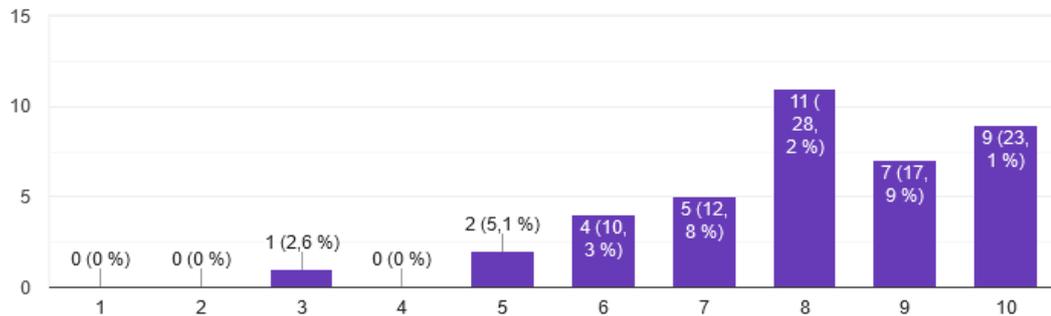


Figura 48. Puntuaciones generales de Blitz Party

Es importante la Figura 48 porque engloba la opinión general de los encuestados e indica que, a pesar de que la mayoría de las puntuaciones son altas, siendo sobre todo de ocho, nueve y diez, hay algunos valores bajos que demuestran que aún hay margen de mejora. Además de todas las preguntas que se acaban de mostrar, se han hecho ciertas cuestiones de respuesta abierta para saber opiniones más extensas de los encuestados. A continuación, exponemos las preguntas y las opiniones más destacables de los encuestados:

- Pregunta abierta uno: ¿Qué es lo que más te ha gustado del juego?
  - Es sencillo aprender a jugar, puede ser divertido para echarse un pique con los amigos.
  - Los minijuegos en sí están muy bien elegidos y son muy entretenidos, sobre todo si buscas superar tus récords.
  - La simpleza, el juego no es super enrevesado y por tanto está bien para jugar de vez en cuando.
- Pregunta abierta dos: ¿Qué es lo que menos te ha gustado?
  - Iniciar sesión cada vez.
  - La dificultad del primer minijuego.
  - El minijuego de disparos es muy fácil.
- Pregunta abierta tres: ¿Tienes alguna sugerencia sobre cómo mejorar el videojuego?
  - Pues en el modo de disparar, que si le das antes de tiempo tendrías que morir, que si no es muy fácil.
  - Rebajad al malo de binky pursuit.
  - Instrucciones para jugar.

Con todos los datos recogidos en mente, el objetivo para el MVP siguiente es mejorar estas puntuaciones mediante la implementación de los cambios mencionados a lo largo de la encuesta y la introducción de funcionalidades nuevas.

### 3.3 Segundo *Minimum Viable Product*

Para este segundo MVP, se ha decidido darle mayor énfasis a la parte multijugador del juego y a lo relacionado con los datos en la nube. Se pueden observar las características seleccionadas en la Figura 49. Además, se ha aprovechado todo el *feedback* recibido durante el MVP anterior para mejorar la funcionalidad de las características que ya están implementadas.

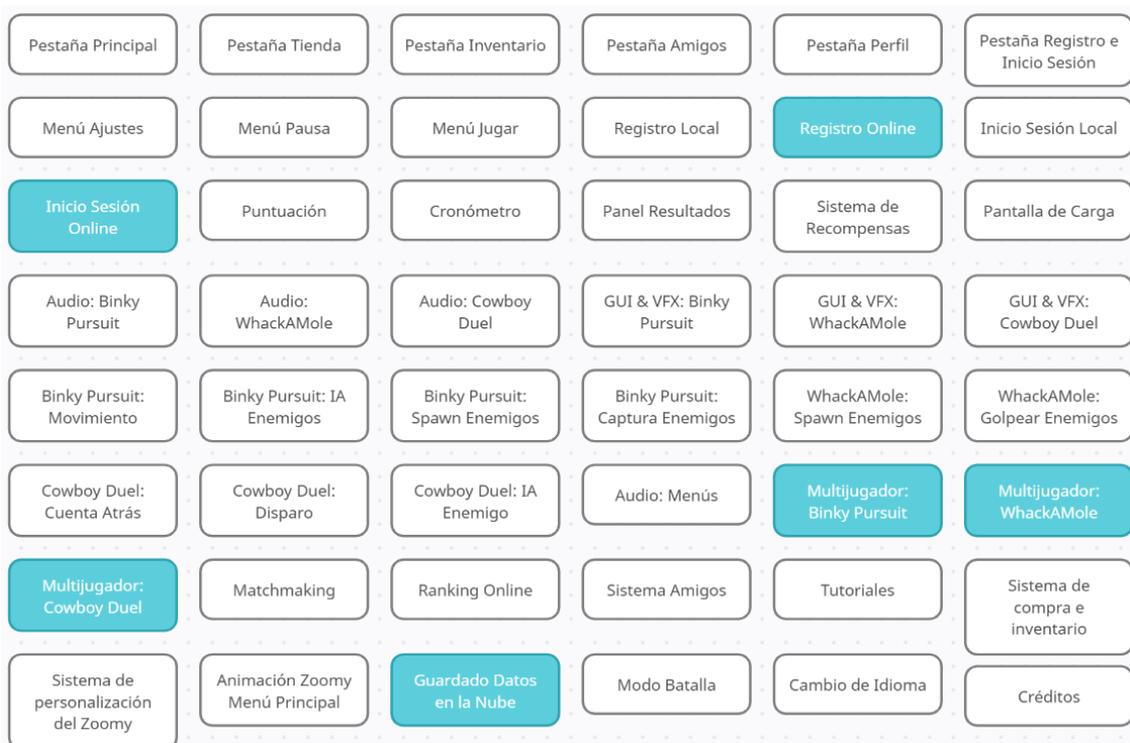


Figura 49. Mapa de características del segundo MVP

Los errores arreglados del MVP anterior son, por ejemplo, que el zorro azul se veía un poco pixelado en *Cowboy Duel* y la pistola se veía demasiado nítida y rompía la inmersión en este minijuego. También, se recibió mucho *feedback* de que el minijuego *Binky Pursuit* era demasiado complicado y por eso se decidió reducir la velocidad de la IA. Otra funcionalidad añadida fue la posibilidad de fallar el disparo en *Cowboy Duel*. Anteriormente, en este minijuego no había penalización por pulsar la pantalla repetidamente. Por eso, se decidió que, si se pulsaba la pantalla demasiado pronto, el zorro fallaría el disparo y se quedaría sin balas para la ronda correspondiente.

Otra novedad de este MVP ha sido la actualización del sistema de registro e inicio de sesión. Anteriormente, esta funcionalidad se realizaba completamente en local del dispositivo y se ha

decidido cambiarlos para que se realicen de manera online haciendo uso de una base de datos en la nube.

### 3.3.1 Desarrollo del MVP

Como hemos comentado en el apartado anterior, el objetivo principal de este MVP ha sido implementar los minijuegos que ya teníamos, pero en su versión multijugador. Para que dos jugadores puedan interactuar dentro del mismo minijuego, se ha utilizado *Mirror*, cuyo funcionamiento se explicará en el capítulo cuatro. Cuando los usuarios seleccionan el modo multijugador, se despliega un menú de búsqueda en la que se espera a que haya dos jugadores disponibles. Una vez formada la pareja de jugadores, empieza el minijuego.

Los tres minijuegos se han podido desarrollar completamente en modo multijugador y su funcionalidad ha sido probada, tanto por el equipo de desarrollo como por los usuarios. A continuación, se explicarán las diferencias existentes entre ambos modos de estos minijuegos.

- ***Binky Pursuit*** (Figura 50): la implementación de este minijuego ha sufrido numerosos cambios, pues el segundo personaje, el zorro azul en este caso, ha pasado a estar controlado por el segundo jugador. Por ello, toda la IA relacionada con su comportamiento ha sido eliminada, y en su lugar el código se ha adaptado para que los dos jugadores dispongan de un joystick y un botón de captura propios. Además, al inicio del minijuego, se han añadido unas flechas que indican cuál es el marcador de cada uno de los jugadores, pues en su defecto el usuario no sabría qué zorro se le ha asignado.



Figura 50. *Binky Pursuit* multijugador

- ***Whack-A-Mole*** (Figura 51): en el modo de un solo jugador de este minijuego únicamente se debía superar una puntuación. No existía un segundo jugador controlado por una IA que podía ser reemplazado por un segundo jugador. Por este motivo, el segundo jugador

también obtiene el control de un martillo de color azul para poder distinguir a ambos jugadores. El funcionamiento del minijuego no ha cambiado, se deben golpear a los topos para conseguir puntos pero, en este caso, solo un jugador puede golpear a un topo y por ello ambos jugadores deben ser rápidos si quieren conseguir batir a su contrincante.



Figura 51. *Whack-A-Mole* multijugador

- **Cowboy Duel:** al igual que en los minijuegos anteriores, unas flechas al inicio de la partida indican qué zorro se le asigna a cada jugador. En el modo para un jugador, el zorro enemigo disparaba aleatoriamente en un intervalo establecido. Ahora, en el modo multijugador, ambos jugadores pueden fallar si clican la pantalla antes de la señal de disparo y, si los dos lo hacen después de esta señal, se proclama vencedor el que lo haya hecho más rápido.

### 3.3.2 Experimento

Al igual que hicimos en el primer MVP, para el segundo hemos realizado otra encuesta ampliando con preguntas sobre las nuevas características añadidas al videojuego. La encuesta se envió a familiares, amigos y conocidos aunque el número de respuestas ha disminuido debido a que, muchos de los encuestados no pueden probar estas nuevas funcionalidades si no disponen de dos dispositivos. Para este experimento se ha decidido incluir solo las preguntas con cambios sustanciales respecto al primer MVP, dado que consideramos que incluirlas todas será redundante.

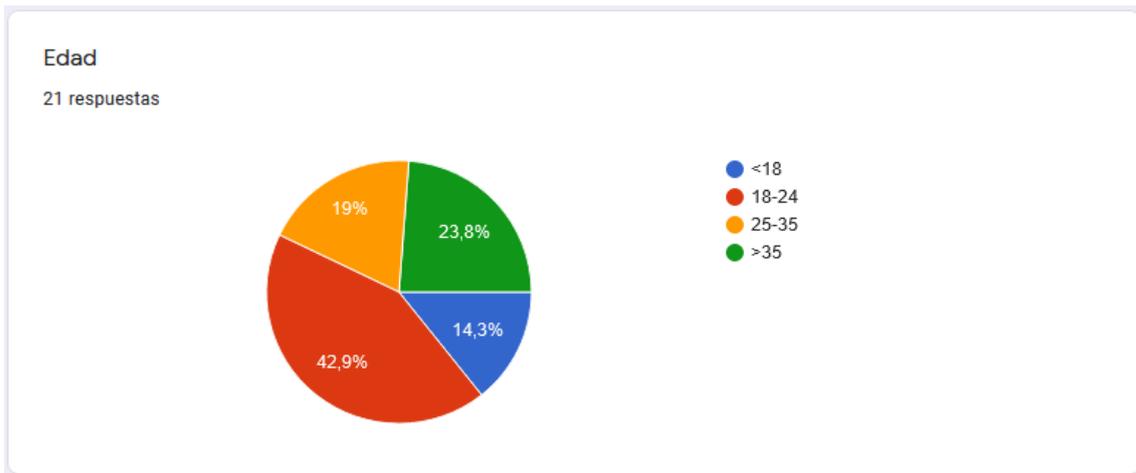


Figura 52. Pregunta rango de edad

Como se puede observar en la Figura 52, el porcentaje de encuestados entre 18 y 24 años se ha reducido respecto a la encuesta realizada en el MVP 1 debido a que esta encuesta se ha dirigido más a familiares y estos no tienen por qué coincidir en rango de edad con los miembros del equipo de desarrollo. Esta reducción de porcentaje explica también el aumento en la cantidad de personas menores de 18 años y entre 25 y 35 años que han respondido la encuesta.

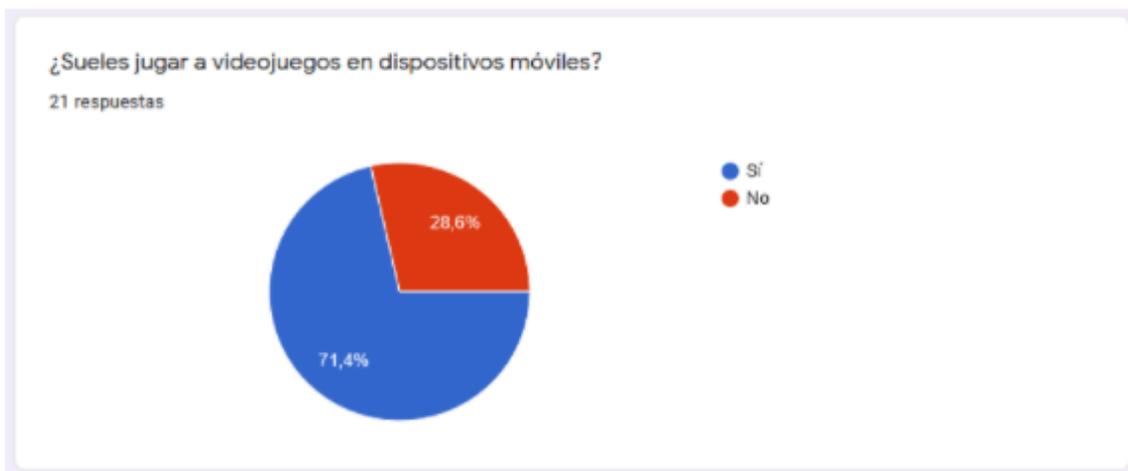


Figura 53. Pregunta jugadores de dispositivos móviles

En la Figura 53 vemos que el porcentaje de jugadores en dispositivos móviles se ha reducido ligeramente, dado que, al ser esta encuesta más localizada en familiares y conocidos, no tienen por qué coincidir en intereses con los miembros del equipo y por tanto juegan menos a videojuegos.

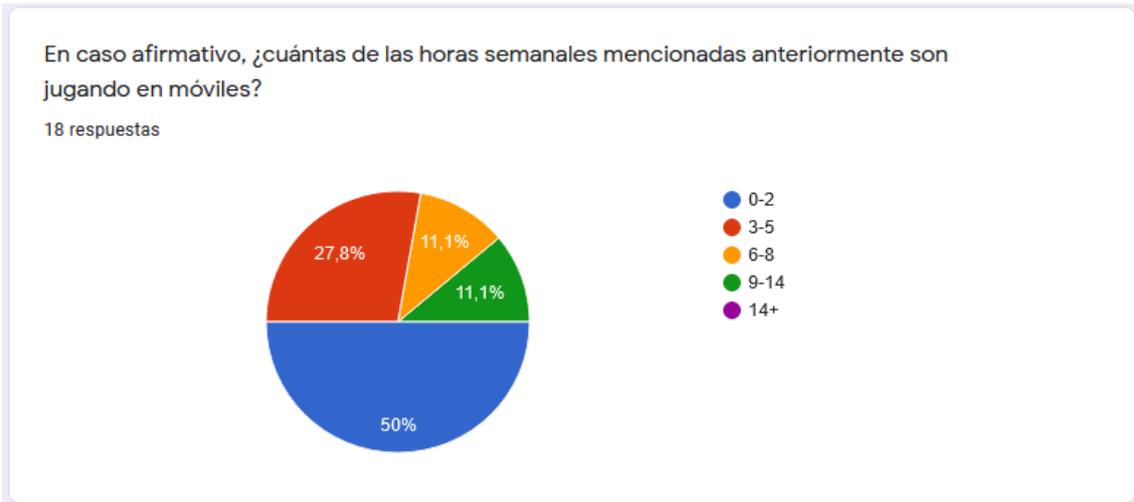


Figura 54. Pregunta horas jugadas en dispositivos móviles

La encuesta anterior (Figura 54) cuadra con lo expresado en la pregunta anterior ya que, los que sí que son jugadores móviles, habitualmente están en un rango de edad en el que no se tiene tanto tiempo para jugar a videojuegos.

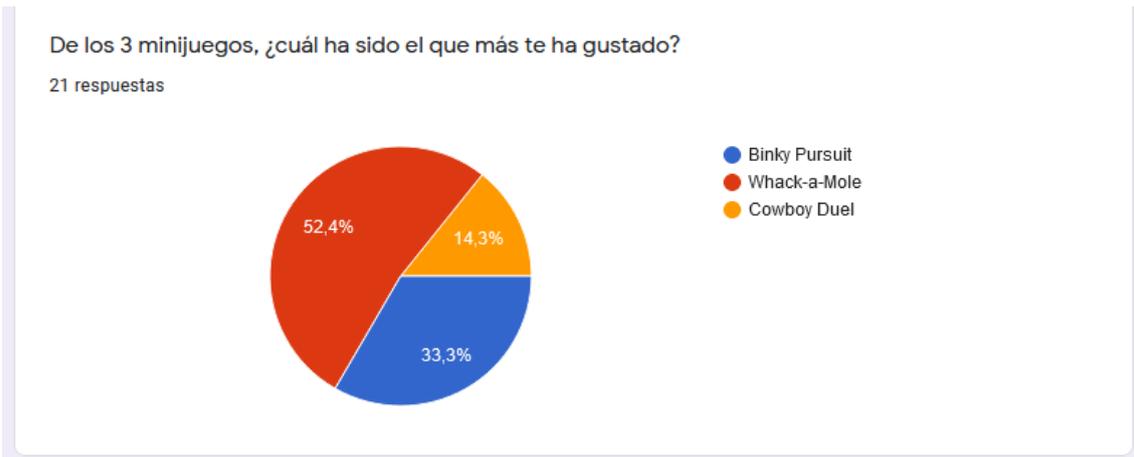


Figura 55. Pregunta minijuego preferido

Si comparamos los datos de la Figura 41 con los de la Figura 55, observamos que *Whack-A-Aole* sigue siendo el minijuego predilecto por los encuestados. Destaca también el crecimiento de *Binky Pursuit*. Esto se puede explicar con que se le ha reducido la dificultad. En el caso de *Cowboy Duel*, el bajo porcentaje de encuestados que lo consideran su favorito puede deberse a la nueva mecánica de fallo del disparo. Esta funcionalidad supone un aumento en la dificultad del minijuego y puede provocar que a los jugadores les guste menos.

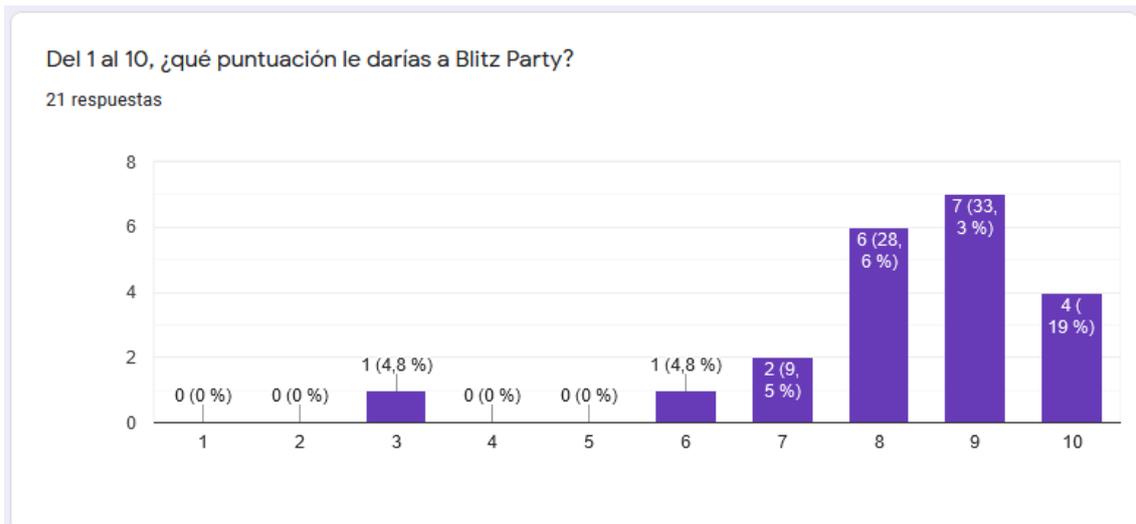


Figura 56. Pregunta puntuación general *Blitz Party*

En las puntuaciones generales del videojuego de la Figura 56, observamos al compararlo con el MVP anterior que generalmente ha aumentado la puntuación de media, se ve que la puntuación de nueve es la más habitual, teniendo también una gran presencia las puntuaciones de ocho y diez.

### 3.4 Publicación del producto y *marketing*

El producto se ha publicado en la *Play Store* de *Google*, obteniendo reseñas altamente positivas como se ve en la Figura 57. Además, se ha obtenido un total de 74 descargas, mayoritariamente proveniente de España pero habiendo algunas también de Estados Unidos, Alemania, Colombia o India como se ve en la Figura 58.

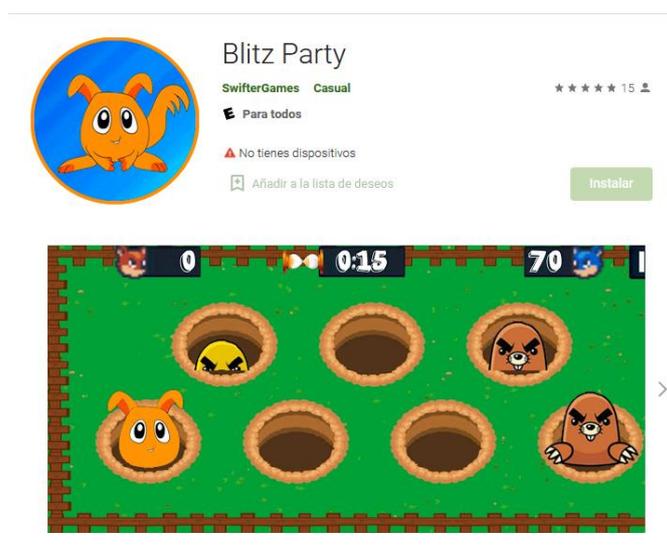


Figura 57. *Blitz Party* en la *Play Store*

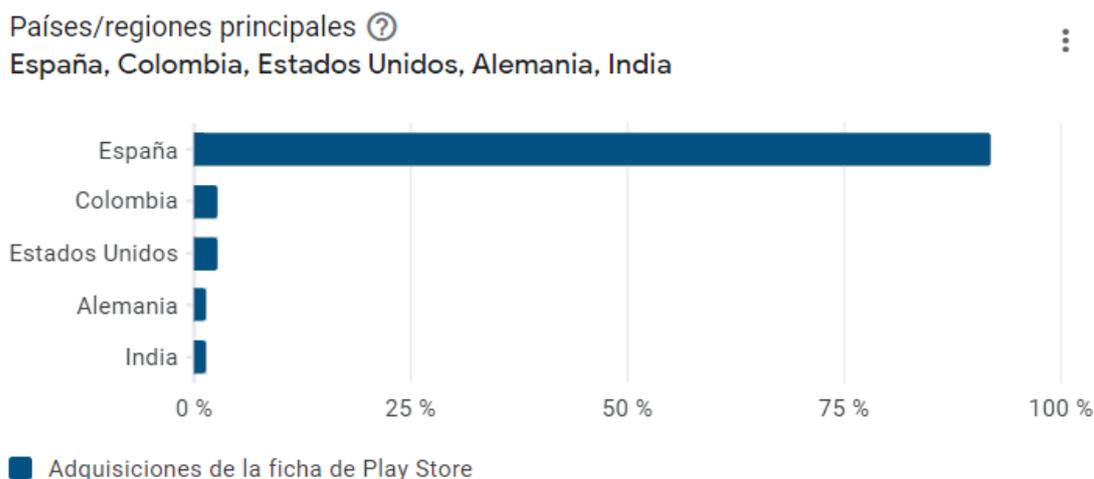


Figura 58. Países desde los que se ha descargado *Blitz Party*

Observamos que hay un total de 15 reseñas con una media total de 5 estrellas. Además, el videojuego aparece simplemente buscando *Blitz Party* en la *Play Store*, lo que le da al juego mayor alcance. Además, para conseguir que este alcance sea aún mayor, se ha hecho *marketing* con las cuentas personales de redes sociales de cada uno de los miembros del equipo y se ha creado una cuenta de *Twitter* con el propósito de promocionar el videojuego, como se ve en la Figura 59.



Figura 59. Cuenta de *Twitter* de *Swifter Games*

## 4. Tecnologías utilizadas

---

En este apartado se describen las herramientas utilizadas en el proyecto así como el porqué de su elección.

### 4.1 Unity Engine



Figura 60. Logo de *Unity Engine*

*Unity* (Figura 60) es un motor de videojuegos multiplataforma propietario de *Unity Technologies*. Su popularidad se debe a su modelo de negocio, que consiste en que cualquier empresa, se trate de un estudio indie o un estudio grande, pueda hacer uso de la herramienta de forma gratuita mientras su proyecto no supere los 100.000 dólares. Sumado a todo esto, la herramienta presenta una curva de aprendizaje menos escarpada en comparación a otros motores similares como *Unreal Engine*. Este último utiliza el lenguaje de programación C++, mientras que *Unity* usa C#, que es el principal y presenta una dificultad menor con respecto a su contraparte de *Unreal*.

En cuanto a su enfoque, *Unreal* está destinado principalmente a videojuegos 3D y por ello no se ajusta a nuestro proyecto en 2D, por lo que *Unity* es la mejor elección por su versatilidad. Para finalizar, la experiencia previa del equipo trabajando con el motor y el lenguaje de programación también ha sido un factor decisivo.

### 4.2 Visual Studio



Figura 61. Logo del IDE *Visual Studio*

*Visual Studio* (Figura 61) es un IDE perteneciente a *Microsoft* que permite desarrollar software con una gran variedad de lenguajes de programación, estando C# y C++ entre los más destacados. La herramienta integra todo tipo de funcionalidades para facilitar el desarrollo, tales como *intellisense* para autocompletar el código y obtener documentación o un *debugger* integrado para depurar y analizar el programa.

La elección de *Visual Studio* se debe a un previo uso del mismo y a su compatibilidad con el lenguaje de programación C# utilizado en el proyecto de *Unity*.

### 4.3 Git/GitHub



Figura 62. Logos de *Git* y *GitHub* respectivamente

*Git* (Figura 62), desarrollado por Linus Torvalds, es sin duda el software de control de versiones más extendido. Esta herramienta permite trabajar en equipo en proyectos y monitorizar cambios en archivos de código fuente.

*GitHub*, haciendo uso de *Git*, es una plataforma de desarrollo software colaborativo que permite a un equipo trabajar paralelamente en un proyecto, alojado en un repositorio en la nube, que puede ser descargado, modificado y actualizado por sus integrantes.

Se ha elegido utilizar *Git* y *GitHub* para que todos los integrantes del equipo puedan gestionar el videojuego de forma cómoda, segura y estable. Esta herramienta facilita el guardado del proyecto en un repositorio en la nube, así como el trabajo en paralelo. Además, ofrece la posibilidad de controlar diferentes versiones del código en caso de que suceda algún conflicto o imprevisto. Otra funcionalidad es la de crear diferentes ramas para trabajar en características nuevas y no probadas sin perjudicar el código de la rama principal, la cual se encuentra en un estado más estable y verificado.



#### 4.4 Worki Process



Figura 63. Logo de *Worki TUNE-UP Process*

*Worki* (Figura 63), de *TUNE-UP Process*, es una herramienta de trabajo utilizada para la realización de proyectos siguiendo metodologías ágiles, permitiendo dividir un proyecto en diferentes Unidades de Trabajo a lo largo de diferentes *sprints*. Esto otorga al equipo mayor facilidad para gestionar la colaboración en paralelo, la distribución de trabajo y la visualización de estadísticas y datos sobre la eficiencia de trabajo.

El equipo ha concurrido en utilizar esta herramienta debido a su uso en diferentes asignaturas a lo largo del transcurso de la rama de Ingeniería del Software, sumado a su comodidad y facilidad.

#### 4.5 Photoshop



Figura 64. Logo de *Adobe Photoshop*

*Adobe Photoshop* (Figura 64), desarrollado por *Adobe Systems Incorporated*, es una herramienta de edición y retoque de imágenes y gráficos.

Se ha decidido el uso de *Photoshop* como editor para cada una de las imágenes y *sprites* utilizadas dentro del videojuego 2D. Los diseños artísticos han sido modificados y ajustados específicamente para cada menú y nivel del juego.

## 4.6 Mirror

*Mirror* (Figura 65) es una API de *networking* de alto nivel para Unity basada en la antigua solución oficial de Unity: *UNet*. Es un proyecto de código *open source* de uso totalmente gratuito y sigue el modelo cliente-servidor, estando ambos integrados en el mismo proyecto. Esto permite crear una *build* del juego *headless* que actúe como servidor y todas las otras *builds* estándares actúen como cliente. En caso de no disponer de un servicio online dedicado para hacer *host* de nuestro servidor, *Mirror* permite que un cliente actúe a su vez de servidor, permitiéndole actuar de anfitrión y crear una conexión *peer-to-peer*.



Figura 65. Logo de *Mirror*

Se ha decidido utilizar Mirror frente a otras librerías por disponer de una documentación más completa y una de las comunidades más grandes de APIs de terceros. Esto, sumado a la retirada de la solución oficial de Unity: *UNet*, que quedó completamente obsoleta con la versión Unity 2018, ha conllevado la búsqueda de otra alternativa, derivando en la decisión de usar Mirror.

## 4.7 Azure PlayFab

*Azure PlayFab* (Figura 66), perteneciente a *Microsoft*, es una plataforma *back-end* que permite añadir y trabajar con funcionalidades online en los videojuegos. *Playfab* ofrece servidores multijugador, *matchmaking*, gestión de datos, analíticas en tiempo real, tablas de clasificación y estadísticas, comunidades online y servicio de chat, entre otras funcionalidades.



Figura 66. Logo de *Azure PlayFab*

Se ha optado por hacer uso de la plataforma de *Microsoft Azure Playfab*, debido a que ofrece un servicio gratuito limitado que satisface las necesidades surgidas durante el desarrollo, facilitando la implementación de características online como registro e inicio de sesión de cuentas en la nube así como almacenaje y gestión de datos internos del videojuego. Del mismo modo, permite hacer uso de sus servidores dedicados hasta un máximo de 750 horas operativas.

# 5. Desarrollo del *back-end*: Análisis

---

## 5.1 Introducción

El *back-end* de una aplicación consiste en líneas generales en la parte de la aplicación que el usuario final no puede ver. Concretamente, se usa esta palabra para referirse sobre todo al guardado de información en bases de datos y a la lógica de la aplicación que procesa la entrada desde el *front-end* y devuelve la salida que se debe mostrar al jugador.

En una arquitectura por capas, el *back-end* constituye la capa de persistencia y es donde se almacenan los datos de los usuarios para mantener la consistencia entre diferentes usos de la aplicación. Esto es especialmente importante en un videojuego dado que, de no existir, cada vez que se iniciase el juego, la aplicación actuaría como si fuera la primera ejecución y, al no existir un sentimiento de progresión, el juego se volvería repetitivo y los jugadores se acabarían cansando.

A la hora de decidir qué funcionalidades era necesario implementar para el *back-end* de la aplicación, se ha decidido usar métodos de elicitación de requisitos. Con esto se ha conseguido seleccionar las características más adecuadas para que los usuarios del videojuego tengan una experiencia satisfactoria con respecto al guardado de datos.

## 5.2 Métodos usados

Se ha decidido usar varios métodos para identificar los requisitos que se implementarán en la aplicación. Estos métodos consisten en:

- *Brainstorming*: El *brainstorming*, también conocido como tormenta de ideas, es una técnica usada en reuniones en grupo que consiste en que los miembros del grupo aporten ideas siendo libres de juicio o crítica. El objetivo del *brainstorming* es generar una gran variedad de ideas, se busca la cantidad en vez de la calidad. El objetivo de esto es que cuantas más ideas se presenten, más posibilidades hay de que existan ideas con proyección de futuro. A partir de aquí, las mejores ideas se seleccionan y se desarrollan.

- Casos de uso: Los casos de uso se usan para identificar y especificar requisitos. Se crea una situación de interacción de un usuario con la aplicación, creando un escenario en el que se asume lo que hace el usuario y se especifica cómo tiene que responder la aplicación a eso. Sirven para obtener una visión completa de cómo tiene que responder la aplicación a escenarios comunes y así conseguir unos requisitos específicos.



### 5.3 Brainstorming

Al principio de la concepción de la idea de negocio, se llevó a cabo un *brainstorming* entre los miembros del equipo de desarrollo. Se generó una gran cantidad de ideas entre las que se seleccionaron las más viables para ser implementadas. En la Figura 67 se muestran todas las ideas generadas en el *brainstorming*, con las ideas relacionadas con el *back-end* marcadas en color verde. Observamos tanto las funcionalidades obvias (como pestaña de registro e inicio de sesión) como también algunas ideas menos comunes como por ejemplo un sistema de recompensas aleatorias.

Dado que esto es un *brainstorming*, no todas las ideas pueden ser implementadas. Por esto, este método de elicitación de requisitos es usado como una guía para decidir qué es lo que finalmente se acaba implementando. De las ideas marcadas en verde en la Figura 67, no todas se pueden acabar añadiendo al videojuego, pero las que no llegan al juego en el presente podrán ser añadidas en un futuro



Figura 67. Brainstorming para Blitz Party

## 5.4 Casos de uso

En la Figura 68 a la Figura 73 se muestran los casos de uso que describen las funcionalidades añadidas a *Blitz Party*.

Nombre:	Registro de usuario
Descripción:	El usuario se registra en la base de datos con un usuario y contraseña
Actores:	Usuario
Precondiciones:	Ninguna
Flujo de eventos:	<ol style="list-style-type: none"> <li>1. El usuario escribe su nombre de usuario en el campo de texto correspondiente.</li> <li>2. El usuario escribe su contraseña en el campo de texto correspondiente.</li> <li>3. El usuario pulsa el botón de <i>Register</i>.</li> <li>4. Se muestra el menú principal con la sesión ya iniciada.</li> </ol>

Figura 68. Caso de uso Registro de usuario

Nombre:	Inicio de sesión de usuario
Descripción:	El usuario entra a su cuenta ya registrada en la aplicación
Actores:	Usuario
Precondiciones:	El usuario tiene una cuenta registrada
Flujo de eventos:	<ol style="list-style-type: none"> <li>1. El usuario escribe su nombre de usuario en el campo de texto correspondiente.</li> <li>2. El usuario escribe su contraseña en el campo de texto correspondiente.</li> <li>3. El usuario pulsa el botón de <i>Sign in</i>.</li> <li>4. Se muestra el menú principal con la sesión ya iniciada y los datos del usuario ya cargados.</li> </ol>

Figura 69. Caso de uso Inicio de sesión de usuario

Nombre:	Ganar <i>acorns</i>
Descripción:	El usuario vence en un minijuego y gana <i>acorns</i> que se guardan en su cuenta
Actores:	Usuario
Precondiciones:	El usuario tiene una cuenta registrada
Flujo de eventos:	<ol style="list-style-type: none"> <li>1. El usuario gana en un minijuego.</li> <li>2. Aparece la pantalla de victoria indicando la cantidad de <i>acorns</i> que ha ganado el usuario.</li> <li>3. El usuario vuelve al menú</li> <li>4. En el menú principal se muestra la cantidad de <i>acorns</i> que posee el usuario teniendo en cuenta los que se acaban de ganar.</li> </ol>

Figura 70. Caso de uso Ganar *acorns*

Nombre:	Guardar volumen de música o efectos de sonido
Descripción:	El usuario cambia el volumen de la música o los efectos de sonido y se guarda para la próxima vez que se inicia sesión
Actores:	Usuario
Precondiciones:	El usuario tiene una cuenta registrada
Flujo de eventos:	<ol style="list-style-type: none"> <li>1. El usuario cambia el volumen de la música o de los efectos de sonido.</li> <li>2. El usuario sale de la aplicación o cierra la sesión.</li> <li>3. El usuario vuelve a iniciar sesión.</li> <li>4. El volumen de la música y de los efectos de sonido es el mismo que había en la última sesión.</li> </ol>

Figura 71. Caso de uso Guardar volumen de música o efectos de sonido

Nombre:	Comprar objetos de la tienda
Descripción:	El usuario compra un objeto en la tienda y ese objeto se guarda en su cuenta
Actores:	Usuario
Precondiciones:	<p>El usuario tiene una cuenta registrada.</p> <p>El usuario tiene suficientes <i>acorns</i> para comprar un objeto.</p>
Flujo de eventos:	<ol style="list-style-type: none"> <li>1. El usuario compra un objeto en la tienda.</li> <li>2. Aparece el cartel de objeto comprado.</li> <li>3. El objeto se añade al inventario.</li> <li>4. La tienda ya no permite comprar el objeto.</li> </ol>

Figura 72. Caso de uso Comprar objetos de la tienda

Nombre:	Personalizar personaje
Descripción:	El usuario equipa uno de los objetos de su inventario a su personaje
Actores:	Usuario
Precondiciones:	El usuario tiene una cuenta registrada.  El usuario tiene objetos en el inventario.
Flujo de eventos:	<ol style="list-style-type: none"> <li>1. El usuario equipa uno de los objetos del inventario</li> <li>2. El objeto se equipa al personaje.</li> <li>3. En el inventario, aparece la opción de desequipar ese objeto.</li> <li>4. El personaje sigue teniendo sus objetos equipados aunque se cierre la sesión y se vuelve a abrir.</li> </ol>

Figura 73. Caso de uso Personalizar personaje

# 6. Desarrollo del *back-end*: Diseño

---

## 6.1 Introducción

El diseño del software es una parte del ciclo de vida de este. En el diseño de software se analizan los requisitos generados anteriormente y se describe la estructura interna del software para facilitar su implementación. El diseño es un proceso iterativo que va mejorando según se va haciendo. Es una parte muy importante del proceso de desarrollo y de él depende en gran parte la calidad del software final.

El proceso de diseño de software puede dividirse en dos partes: diseño preliminar y diseño detallado.

- El diseño preliminar se centra en traducir los requisitos generados durante el análisis de software a la arquitectura del software.

- El diseño detallado se centra en crear una estructura arquitectónica refinada.

Algunos métodos de diseño que se han aplicado en este proyecto son:

- Patrones: los patrones de diseño son soluciones a problemas de software. Uno de los objetivos de los patrones es evitar tener que buscar nuevas formas de solucionar problemas ya conocidos. Un patrón se puede aplicar para resolver un mismo caso en situaciones diferentes. El diseñador debe ser el que decida, en cada contexto, si un patrón es aplicable o no.

- Diagrama de clases: un diagrama de clases en Lenguaje Unificado de Modelado (UML) es un diagrama que muestra las clases del sistema, sus atributos y las relaciones entre objetos. Sirve para visualizar estas relaciones y atributos y, por tanto, facilitar su implementación.



## 6.2 Patrones

En el diseño del *back-end*, se ha decidido utilizar el patrón Fachada. Este patrón, como se observa en la Figura 74, funciona de forma que se crea una interfaz simple (*ModuleA*) para comunicarse con un subsistema complejo (*ModuleB* y *ModuleC*).

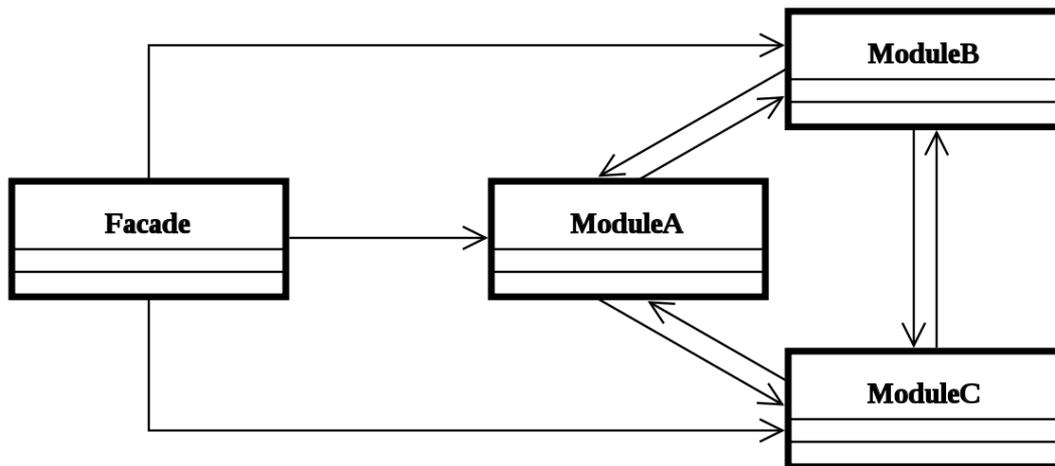


Figura 74. Esquema de funcionamiento del patrón Fachada

En el caso de *Blitz Party*, se creará una clase que se comunicará con las diferentes clases usadas para guardar información en la base de datos, simplificando así este proceso y eliminando la necesidad de que en cada momento haya que comunicarse con una clase diferente. Esta centralización de las comunicaciones con la base de datos reduce la probabilidad de que se produzcan errores a la hora de escribir código y por tanto reduce la cantidad de trabajo de refactorización que es necesario hacer.

### 6.3 Diagrama de clases

Para el diseño del *back-end*, se ha creado un diagrama de clases hecho con UML que se ve en la Figura 75.



Figura 75. Diagrama de clases de *Blitz Party*

Como vemos en la Figura 75, el sistema consta de un usuario con sus datos, siendo estos un identificador, un nombre de usuario, una contraseña para iniciar sesión, la cantidad de *acorns* que posee, el volumen actual al que tiene el ajuste de la música y el volumen al que tiene el ajuste de los efectos de sonido. Cada usuario además posee un único inventario y un único *Zoomy*.

El inventario posee los objetos que se han comprado anteriormente de la tienda. Estos objetos tienen un precio en *acorns* que es lo que cuesta comprarlos de la tienda y una parte del cuerpo, que indica dónde los llevará el *Zoomy* cuando sean equipados. El *Zoomy* puede equipar hasta tres objetos, uno en los pies, uno en el cuerpo y uno en la cabeza. Todos estos objetos están disponibles para comprar en una tienda.



# 7. Desarrollo del *back-end*: Implementación

---

## 7.1 Implementación del *back-end* local

Para el guardado local de información de los jugadores se ha decidido usar la clase *PlayerPrefs* de *UnityEngine*. Esta clase guarda los ajustes de los usuarios entre distintas sesiones de juego. Se ha creado una clase *Database* haciendo uso de *PlayerPrefs*. Esta se ha usado principalmente para mantener la cantidad de *acorns* que tiene el jugador, las estadísticas de partidas jugadas y victorias conseguidas, su nombre de usuario y contraseña y los objetos que se han comprado y que se tienen el inventario.

### Static Methods

<a href="#">DeleteAll</a>	Removes all keys and values from the preferences. Use with caution.
<a href="#">DeleteKey</a>	Removes the given key from the PlayerPrefs. If the key does not exist, DeleteKey has no impact.
<a href="#">GetFloat</a>	Returns the value corresponding to key in the preference file if it exists.
<a href="#">GetInt</a>	Returns the value corresponding to key in the preference file if it exists.
<a href="#">GetString</a>	Returns the value corresponding to key in the preference file if it exists.

Figura 76. Métodos estáticos de la clase *PlayerPrefs* (1)

<a href="#">HasKey</a>	Returns true if the given key exists in PlayerPrefs, otherwise returns false.
<a href="#">Save</a>	Writes all modified preferences to disk.
<a href="#">SetFloat</a>	Sets the float value of the preference identified by the given key. You can use PlayerPrefs.GetFloat to retrieve this value.
<a href="#">SetInt</a>	Sets a single integer value for the preference identified by the given key. You can use PlayerPrefs.GetInt to retrieve this value.
<a href="#">SetString</a>	Sets a single string value for the preference identified by the given key. You can use PlayerPrefs.GetString to retrieve this value.

Figura 77. Métodos estáticos de la clase *PlayerPrefs* (2)

En la Figura 76 y la Figura 77 se observan los métodos estáticos de la clase *PlayerPrefs*. Estos métodos funcionan como un diccionario en el que se guardan claves y a cada clave se le asocia un valor. Los métodos *Set* funcionan de forma que se les da una clave y un valor y se guarda el valor asociado a la clave mientras que a los métodos *Get* se les pasa una clave y devuelven el valor asociado a ella

Se han usado los métodos estáticos mencionados anteriormente para guardar y acceder a datos. Además, en la clase *Database* se han implementado métodos propios para adaptar la

funcionalidad de *PlayerPrefs* a lo que se necesitaba para este proyecto. Por ejemplo se usa esto para guardar tanto el número total de objetos que posee el usuario como qué objetos concretos son.

A esto se le ha añadido una función de inicio de sesión (Figura 78) por medio de la cual cada jugador ha de introducir su nombre de usuario antes del inicio de cada sesión de juego, creando así un perfil de usuario al que se le asocian los datos almacenados. Además, la clase *Database* se ha acompañado de una clase *User*, que crea un objeto cuando un usuario nuevo inicia sesión mediante el método *AddUser()* de la clase *Database* y accede a este objeto las siguientes veces que este usuario juegue. La clase *User* se usa para tener objetos de usuarios y así facilitar el guardado y carga de sus datos de estadísticas, objetos comprados o *acorns* conseguidos, entre otras.



Figura 78. Pantalla de inicio de sesión *offline*

## 7.2 Implementación del *back-end* en la nube

El guardado de datos en la nube se ha efectuado usando *Azure PlayFab*. Se ha instalado en *Unity* el paquete: *PlayFab Unity Editor Extensions Asset Package* como observamos en la Figura 79 para extender las funciones del editor como se ve en la Figura 80 y también el SDK de *PlayFab* desde el editor de *Unity* para poder efectuar llamadas a la API.

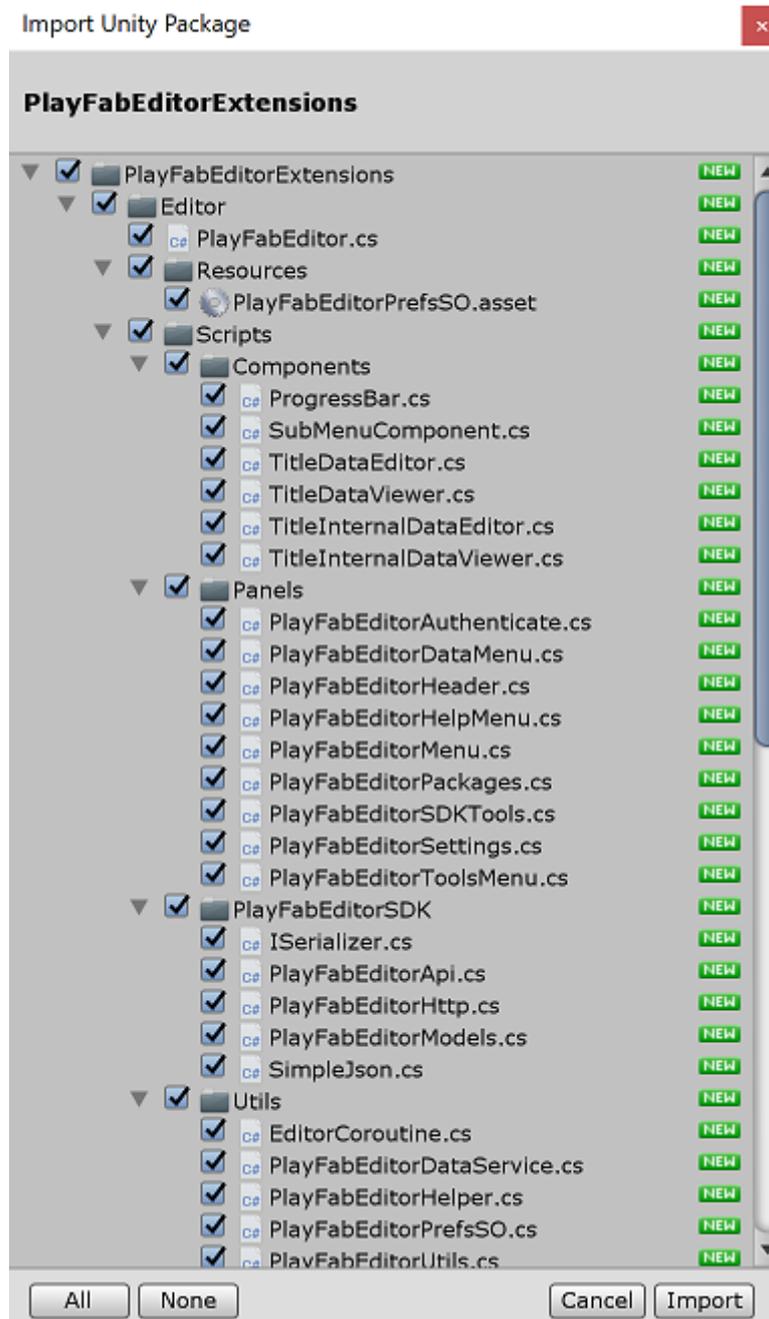


Figura 79. Importación del paquete de *Azure PlayFab*

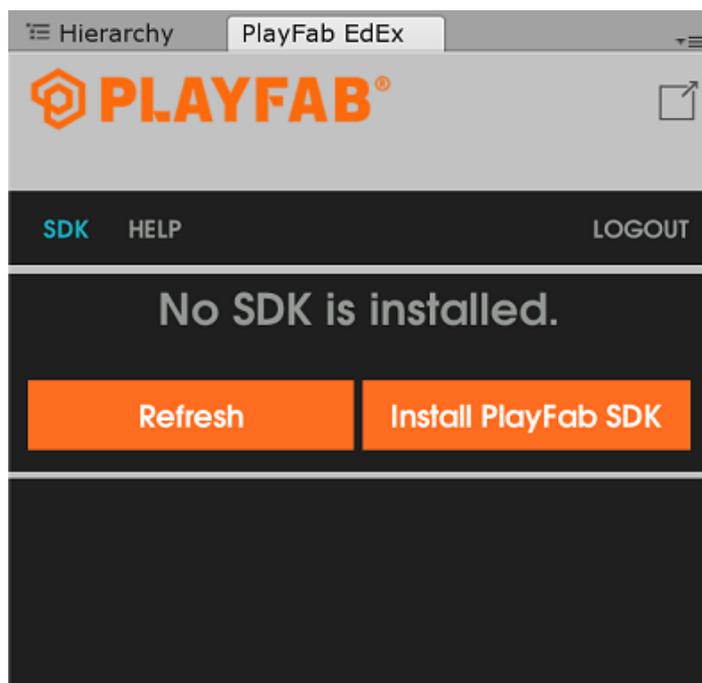


Figura 80. Editor de *Unity* tras la instalación del paquete *PlayFab*

Tras instalar el SDK, la ventana de *PlayFab* en el editor cambia y se pueden modificar ajustes sobre el juego. A partir de este momento, la extensión del editor se ve como en la Figura 81 y se puede usar *PlayFab* para guardar datos en la nube.

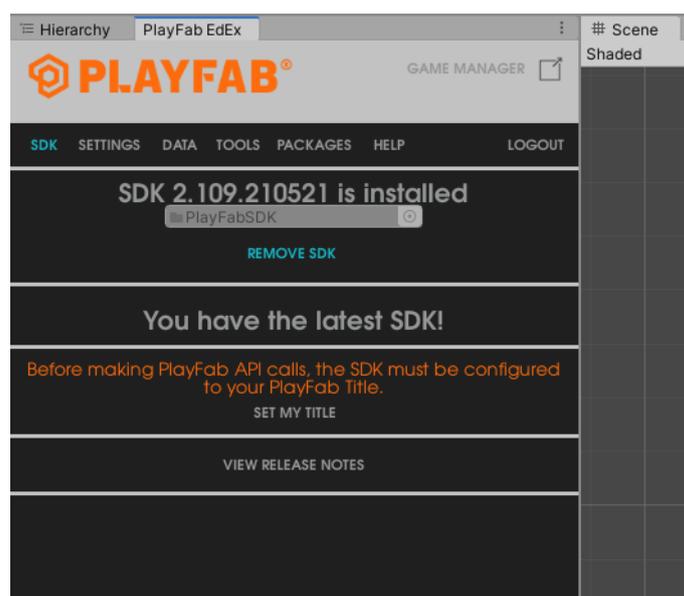


Figura 81. Editor de *Unity* tras instalar el SDK de *PlayFab*

En el primer MVP, dado que el guardado de datos se implementaba de forma local por dispositivo, obviamente estos datos no estaban disponibles al cambiar de dispositivo. Por eso, se ha decidido para el segundo MVP hacer estos datos accesibles desde cualquier dispositivo asociándolos a una cuenta de usuario que se crea al iniciar sesión. Para esto se ha usado *Azure PlayFab*, se han creado clases para implementar sus funcionalidades y se ha adaptado la ventana de inicio de sesión.



Figura 82. Pantalla de inicio de sesión *online*

Como vemos en la Figura 82, ahora se necesita introducir un nombre de usuario y una contraseña. Con estos datos se registra al usuario en el servidor de *PlayFab* y se puede iniciar sesión en cualquier dispositivo introduciendo el nombre de usuario y la contraseña. Se ha implementado esto con un *script* llamado *PlayFabLogin* que guarda los datos del usuario en la base de datos

local y cada cierto tiempo fijado, al cerrar sesión o cerrar la aplicación esos datos se guardan en la nube. Las contraseñas se mandan directamente a *PlayFab*, donde se guardan siendo encriptadas por medio de un *hash* (que es un algoritmo de encriptación irreversible) y *salting*, que consiste en agregar un valor aleatorio único para cada usuario al algoritmo de encriptación *hash*, que hace que los valores que devuelve este algoritmo sean diferentes para cada usuario[6].

Esto funciona de forma que se guardan los datos en la nube cada un intervalo de diez minutos, en el cierre de sesión o en el cierre de la aplicación. Cada vez que esto ocurre, se produce una llamada al método *StoreOnCloudUserData()* en dos de los casos mencionados anteriormente (el del cierre de sesión se hace en el mismo *script* en que se cierra la propia sesión) y este método produce una subsiguiente llamada al método *SetUserData()* de la clase *CloudStoragePlayFab*, que efectúa la subida de los datos a la nube.

Se ha decidido implementar el guardado de datos de forma periódica o al cierre porque guardar datos en la nube necesita un tiempo mayor que guardarlos localmente y hacerlo cuando se produce cualquier cambio, como se hacía en el MVP uno, sería poco práctico. Por tanto, esto hace que para el usuario la aplicación tenga el mismo rendimiento que en el MVP anterior y dado que cubrimos el cierre de sesión y el cierre de aplicación, no hay ningún caso en el que estos datos no queden guardados.

La clase *CloudStoragePlayFab* es la que efectúa la subida o bajada de datos de la nube. Concretamente, se guarda un id de *PlayFab*, el nombre de usuario, el número de *acorns* que se tienen y el volumen de la música y de los efectos de sonido. En el futuro, el objetivo es guardar en la nube todos los datos de los usuarios, incluyendo los objetos comprados y equipados y las estadísticas de partidas jugadas y ganadas.

El método *StoreUserDataOnCloud()* mencionado anteriormente, extrae de la base de datos local el id y el nombre de usuario del jugador y guarda en la nube los *acorns*, el volumen de la música y de los efectos de sonido. Además, efectúa una llamada al método *SetUserData()* de la clase *CloudStoragePlayFab* para guardar estos datos en la nube.

Además, una de las quejas recibidas en la encuesta que acompañaba al primer MVP fue: “es muy molesto iniciar sesión cada vez que entras al juego”. El equipo de desarrollo estuvo de acuerdo con esto y decidió que, si se iniciaba sesión en un dispositivo, se recordaría esto para las siguientes sesiones de juego y no sería necesario volver a iniciar sesión. Para esto, se usó un *script* sencillo cuya función es comprobar si se ha iniciado sesión anteriormente. Esto se hace guardando localmente un *booleano* que indica si la sesión se ha iniciado. En el caso de ser verdadero se pasará directamente al menú principal las siguientes veces que se inicie la aplicación, mientras que en el caso contrario se iniciará con la ventana de iniciar sesión.



Además, como se ve en la Figura 83 ha sido necesario modificar el menú de ajustes para añadir un botón de cerrar sesión ya que, de no hacerlo, no sería posible iniciar sesión con otros usuarios en un dispositivo. Al cerrar sesión con este botón se aplicará también a la comprobación mencionada anteriormente y las siguientes veces se iniciará la aplicación en la ventana de inicio de sesión o con el nuevo usuario en caso de haberlo.

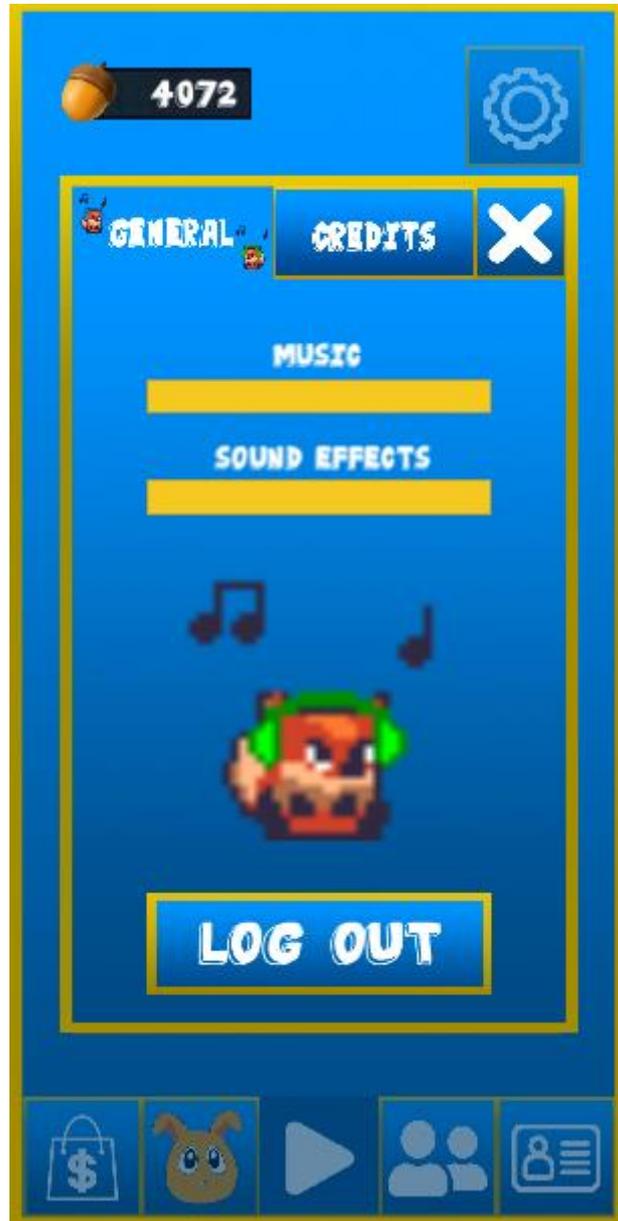


Figura 83. Menú de ajustes con cierre de sesión

# 8. Desarrollo del back-end: Resultados

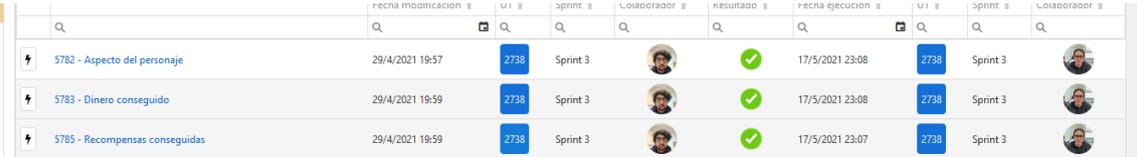
## 8.1 Introducción

Al finalizar la implementación de las funcionalidades de *software*, estas han de ser probadas para comprobar, no solo que funcionan correctamente, sino que, además, cumplen los requisitos definidos en la parte de análisis. Para esto, normalmente las funcionalidades se prueban por un equipo especializado. Sin embargo, dado que este es un proyecto de emprendimiento y no se dispone de este equipo, se ha decidido recurrir a otros métodos:

- Pruebas de aceptación (PA): se han realizado pruebas de aceptación por parte de otros miembros de este equipo de desarrollo. Se han llevado a cabo por medio de la herramienta *Worki*. Estas pruebas se pueden pasar, no pasar o pasar con ciertos apuntes. Cuando una prueba no se pasa, la funcionalidad correspondiente es devuelta a la parte de programación y se arreglan los fallos que tenga.
- Encuestas a usuarios: una vez ya se ha comprobado que todo funciona correctamente, también se quiere saber si es cómodo para los usuarios y coincide con lo que ellos prefieren. Así, por medio de encuestas, se puede saber cuánto les gusta a los usuarios el modo en el que funcionan ciertas características.

## 8.2 Pruebas de aceptación

Se han realizado pruebas de aceptación para todas las Unidades de Trabajo (UT) que estaban registradas en *Worki*.



	fecha modificacion	U.T.	sprint	Colaborador	resultado	fecha ejecucion	U.T.	sprint	Colaborador
5782 - Aspecto del personaje	29/4/2021 19:57	2738	Sprint 3		✓	17/5/2021 23:08	2738	Sprint 3	
5783 - Dinero conseguido	29/4/2021 19:59	2738	Sprint 3		✓	17/5/2021 23:08	2738	Sprint 3	
5785 - Recompensas conseguidas	29/4/2021 19:59	2738	Sprint 3		✓	17/5/2021 23:07	2738	Sprint 3	

Figura 84. Prueba de aceptación para Guardado local de datos sobre el usuario

En la Figura 84 observamos que, para la UT de Guardado local de datos sobre el usuario, se han pasado las tres pruebas de aceptación. Estas pruebas de aceptación comprueban que se guarde correctamente el aspecto del personaje, el dinero que se ha conseguido y también las recompensas que se han conseguido, es decir, los objetos que se han comprado.



	Fecha modificación	UI	Sprint	Colaborador	Resultado	Fecha ejecución	UI	Sprint	Colaborador
5776 - Login	29/4/2021 19:43	2726	Sprint 3		✓	16/5/2021 20:00	2726	Sprint 3	
5778 - Dinero	29/4/2021 19:51	2726	Sprint 3		✓	16/5/2021 20:00	2726	Sprint 3	
5779 - Número de victorias	29/4/2021 19:52	2726	Sprint 3		✓	16/5/2021 19:59	2726	Sprint 3	

Figura 85. Prueba de aceptación para *User profile*

Vemos en la Figura 85 que todas las PA asociadas a la UT de *User profile* han pasado. La PA de *Login* comprueba que el usuario pueda iniciar sesión correctamente y su nombre se muestre en la pestaña de estadísticas. La PA de *Dinero* comprueba que en el menú se muestra correctamente el dinero que el usuario tiene, mientras que la PA *Número de victorias* comprueba que en la pestaña de estadísticas se muestren correctamente las victorias que el jugador ha conseguido en cada minijuego.

	Fecha modificación	UI	Sprint	Colaborador	Resultado	Fecha ejecución	UI	Sprint	Colaborador
5847 - El usuario se logea correctamente	26/5/2021 18:58	2776	Sprint 4		✓	25/6/2021 12:29	2776	Sprint 4	
5848 - El usuario se registra correctamente	26/5/2021 19:00	2776	Sprint 4		✓	25/6/2021 12:28	2776	Sprint 4	

Figura 86. Pruebas de aceptación para *Online login and Register Functionality*

Para *Online login and Register Functionality*, como vemos en la Figura 86, las dos pruebas de aceptación han pasado también correctamente. Esta unidad de trabajo consiste en el inicio de sesión y el registro correctos una vez se ha pasado esta funcionalidad a online. Las dos pruebas de aceptación que se observan consisten en el inicio de sesión y el registro en la base de datos *online*.

	Fecha modificación	UI	Sprint	Colaborador	Resultado	Fecha ejecución	UI	Sprint	Colaborador
5843 - Guardado de datos correcto	26/5/2021 18:41	2777	Sprint 4		✓	25/6/2021 11:53	2777	Sprint 4	
5844 - Cargado de datos correcto	26/5/2021 18:43	2777	Sprint 4		✓	25/6/2021 11:53	2777	Sprint 4	

Figura 87. Pruebas de aceptación para *Data Persistence on the Cloud*

Se observa en la Figura 87 que se han pasado las dos pruebas de aceptación para la unidad de trabajo *Data Persistence on the Cloud*. Esta unidad de trabajo consiste en que los datos de los usuarios se guarden en la nube y la siguiente vez que estos usuarios inicien sesión se carguen a su cuenta.

### 8.3 Encuestas a usuarios

Además de las pruebas de aceptación, se han realizado encuestas a usuarios para comprobar su satisfacción con el guardado de datos de *Blitz Party*. Se han realizado dos preguntas, una sobre la cantidad de estadísticas guardadas y la otra sobre el inicio de sesión y la posibilidad de incluirlo con cuentas de terceros.

¿Las estadísticas guardadas que se muestran en su pestaña te parecen suficientes o guardarías más?

21 respuestas

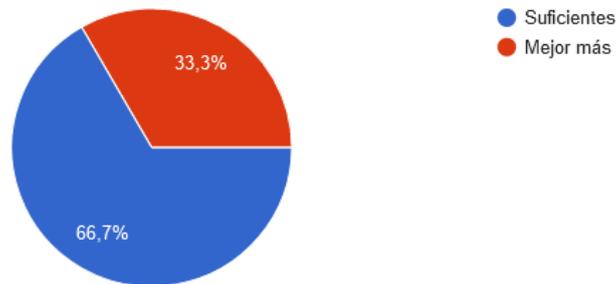


Figura 88. Pregunta sobre las estadísticas que se guardan

En la Figura 88 vemos que la mayoría de las personas que han respondido a la encuesta opinan que con las estadísticas que se guardan actualmente es suficiente. Estas estadísticas, que consisten en el número de partidas jugadas y el número de victorias ofrecen bastante información sobre el historial de juego de cada minijuego.

De todas formas, esta respuesta no cierra completamente la puerta a que se añadan nuevas estadísticas en un futuro, como puede ser el tiempo dedicado a jugar a cada minijuego o el tiempo de juego total. También se pueden añadir en un futuro las mismas estadísticas, pero diferenciando entre modo multijugador o individual.

Otra opción es añadir estadísticas concretas de cada minijuego como pueden ser por ejemplo el número de *binkies* capturados en total, el número de topes de cada tipo golpeados en *Whack-A-Mole* o el tiempo de disparo medio en *Cowboy Duel*. Sería también interesante añadir récords a nivel global para fomentar la competitividad y que los jugadores quieran superarse unos a otros.



¿Te gustaría un inicio de sesión con cuentas de terceros o te parece bien el que hay actualmente?

21 respuestas

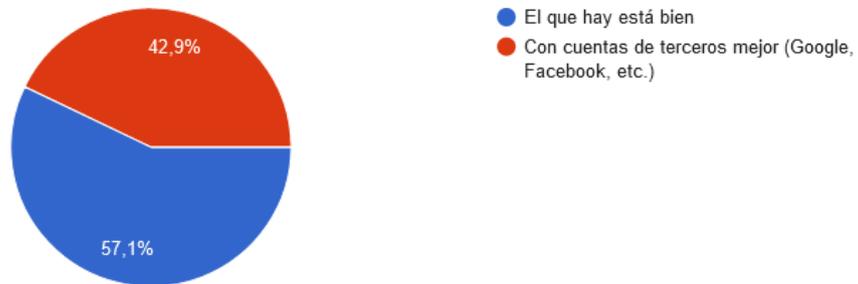


Figura 89. Pregunta sobre inicio de sesión con cuentas de terceros

Se observa en la Figura 89 que la mayoría de encuestados opinan que el inicio de sesión actual está bien. Sin embargo, teniendo en cuenta el escaso margen con el que se ha obtenido esta respuesta y que el inicio de sesión con cuentas de terceros es una característica cómoda, se va a hacer como en las estadísticas de la pregunta anterior y dejar la puerta abierta a una posible adición de esta funcionalidad en el futuro.

## 9. Trabajo adicional

Además del trabajo realizado en la parte del *back-end* del proyecto, se han hecho otras aportaciones y se ha ayudado a los demás miembros del equipo de desarrollo en sus respectivas partes.

En el minijuego *Binky Pursuit*, se ha ideado e implementado el movimiento de los *binkies* mediante el uso de *pathfinding*. En el minijuego *Whack-A-Mole*, se ha implementado el ratio de aparición de los enemigos. Por último, en el minijuego *Cowboy Duel*, se han hecho todas las animaciones y se han implementado para que respondan a los eventos que ocurren en el minijuego.

### 9.1 Pathfinding en Binky Pursuit

En *Binky Pursuit*, los *binkies* salen de los agujeros y corren por el mapa. Se decidió que, para conseguir una sensación de caos, los movimientos de los *binkies* serían aleatorios por el mapa.



Figura 90. Objeto *pathfinder* de *Unity*.

Para hacer posible el cálculo de estos movimientos, se ha añadido a la escena de *Unity* un objeto *pathfinder*, del paquete *A\* Pathfinding*. Este objeto consiste en la cuadrícula que se ve en la Figura 90. Se ha decidido hacer la cuadrícula más pequeña que el mapa porque los *binkies* tienen inercia al cambiar de dirección y con este tamaño de cuadrícula tienen espacio para no salir del mapa.

Los *binkies*, al salir de cualquier *spawner* (los agujeros marcados con el símbolo morado), calculan un punto aleatorio de dentro de esta cuadrícula y se dirigen hacia él. Al colisionar con uno de los agujeros, el *binky* se esconde y al colisionar con uno de los personajes jugadores, se

activa la posibilidad de que los jugadores pulse el botón de captura y si lo hace, el *binky* también desaparece.

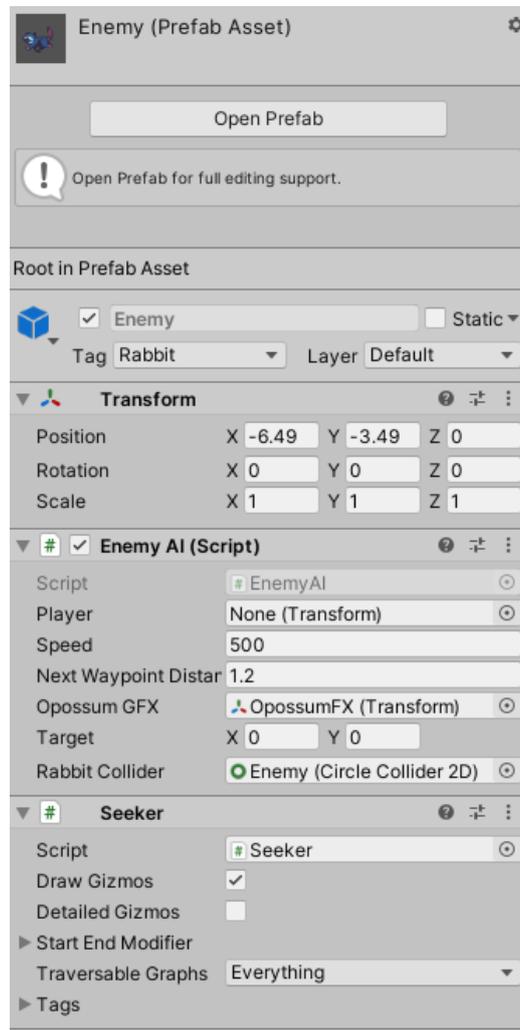


Figura 91. Prefab de *binky* para *Binky Pursuit*.

La Figura 91 muestra el *prefab* de un *binky*, vemos que tiene un componente *Seeker*, que calcula el siguiente punto al que se desplazará el *binky*. El otro componente, *Enemy AI*, es en el que se configura la velocidad de los *binkies* (se consideró que 500 era una velocidad óptima para que se pudieran atrapar, pero no con demasiada facilidad). Además, en este mismo componente se configura que cuando un *binky* se encuentre a 1.2 unidades de distancia del punto al que se dirige, se calculará el siguiente punto.



Figura 92. Movimiento de binkies en Binky Pursuit

En la Figura 92 observamos los *binkies* dirigiéndose a diferentes puntos del mapa con los *gizmos* de las líneas verdes marcando a dónde van y los de los cuadrados rojos marcando la zona de los agujeros que les harán esconderse. Estos *gizmos* son ayudas visuales que se pueden añadir a una escena de *Unity* para facilitar la depuración durante el desarrollo de un juego.

## 9.2 Aparición de enemigos en Whack-A-Mole

En el minijuego *Whack-A-Mole* se ha colaborado con un compañero de equipo para implementar el *spawn* de los diferentes tipos de enemigos.



Figura 93. Mapa de *Whack-a-Mole*

En la Figura 93 vemos el mapa de *Whack-A-Mole*. Desde cada uno de los agujeros pueden salir topos, topos dorados o *zoomies* que darán o quitarán puntos al jugador. Un topo normal otorga un punto, un topo dorado otorga cinco puntos mientras que un *zoomy* resta tres puntos.

Esto se ha implementado de forma que se tiene un *array* de booleanos, cada uno de ellos correspondiente a uno de los agujeros. Con este *array* se selecciona uno de los agujeros aleatoriamente y se comprueba que no esté ocupado. En caso de estar ocupado, se busca aleatoriamente otro agujero hasta encontrar uno que esté libre.

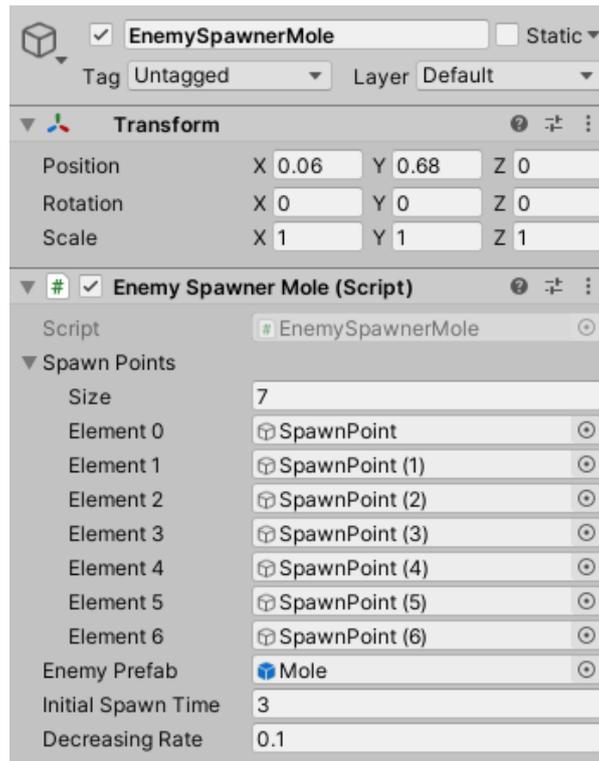


Figura 94. *Spawner* de topos

En la Figura 94 vemos el objeto encargado de hacer aparecer los topos. Vemos que guarda los siete *spawn points* en un *array*. Además, es necesario destacar los atributos *Initial Spawn Time* y *Decreasing Rate*. Estos dos atributos controlan el momento en el que aparecen los topos. Funciona de forma que el primer topo aparecerá a tres segundos del inicio de la partida y el intervalo de aparición se reducirá en 0,1 segundos cada vez que aparezca uno. Estos valores son de 20 y cinco para el topo dorado y de cinco y 0,2 para el *zoomy*. Se ha añadido el *decreasing rate* para que la partida empiece con un ritmo más lento y vaya cogiendo velocidad y frenetismo según avanza el tiempo. También se ha añadido un límite mínimo para la velocidad de aparición de cada enemigo para evitar así posibles *bugs*.

### 9.3 Animaciones en Cowboy Duel

También se han diseñado y realizado todas las animaciones para el minijuego *Cowboy Duel*. Por la naturaleza de este minijuego, las animaciones son lo que decide el estado actual de la partida y les da pistas a los jugadores sobre lo que va a pasar a continuación.



Figura 95. Animación *Idle* de *Cowboy Duel*

En la Figura 95 está la animación que se ve al empezar el minijuego, cuando los zorros están mirándose el uno al otro en el centro de la pantalla. En esta animación el zorro cambia unas cuantas veces de *sprite idle*, de forma que da la impresión de que está respirando lentamente.



Figura 96. Animación *Run* de *Cowboy Duel*

La Figura 96 es la animación que los zorros hacen cuando se desplazan desde el centro a los extremos de la pantalla y se quedan orientados hacia fuera de esta. En esta animación, los dos zorros se mueven a la vez hacia el extremo de la pantalla, como en los clásicos duelos del oeste.

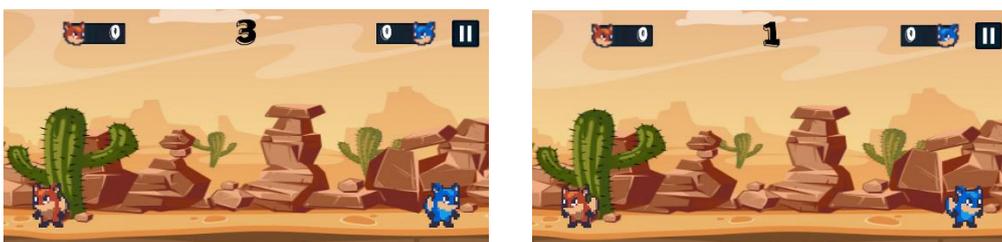


Figura 97. Animación *Wait to shoot* de *Cowboy Duel*

En la Figura 97 se observa la animación que se ejecuta durante la cuenta atrás y mientras los zorros esperan a que aparezca la señal de disparo. Esta animación es parecida a la *idle*, los zorros respiran, pero en esta ocasión más lentamente, dando mayor sensación de tensión dado que están a punto de disparar.



Figura 98. Animación *shoot* de *Cowboy Duel*

En la Figura 98 vemos la animación de disparo, que se ejecuta cuando un jugador pulsa la pantalla después de que haya aparecido la señal de disparo. En esta animación el zorro se gira, saca la pistola y dispara. En caso de que el jugador pulsara la pantalla antes de que aparezca el botón de disparo, la animación que se ejecutará será una versión ligeramente modificada de esta, en la que el zorro dispara hacia el cielo.



Figura 99. Animación *death* de *Cowboy Duel*

Observamos en la Figura 99 la animación de muerte, que se ejecutará cuando el jugador haya disparado después del otro. Esta animación alterna entre los dos *sprites* de muerte del zorro.

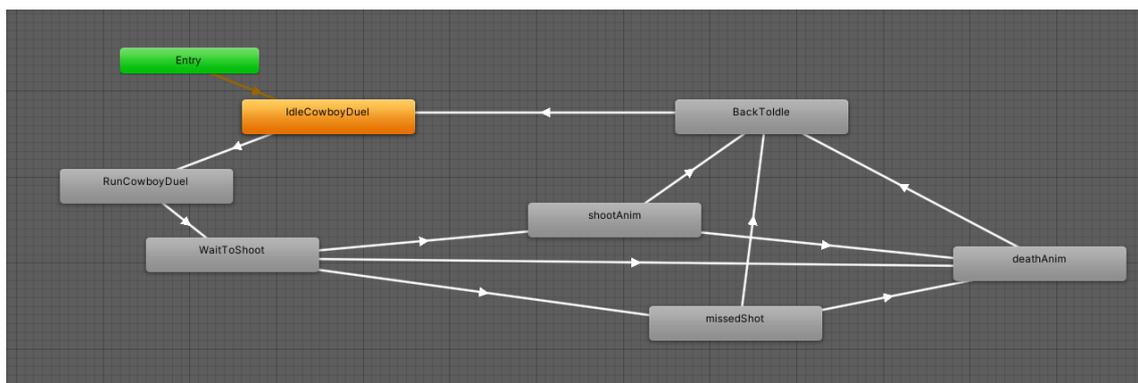


Figura 100. Máquina de estados de *Cowboy Duel*

La Figura 100 muestra la máquina de estados que controla el funcionamiento del minijuego de *Cowboy Duel*. Se empieza en el estado *IdleCowboyDuel* y se llega por tiempo hasta *WaitToShoot*, pasando por *RunCowboyDuel*. Al llegar a *WaitToShoot* se puede ir a varios estados dependiendo del *trigger* que se dispare. Si el jugador dispara antes que su rival, se ejecutará *shootAnim*, si dispara después se ejecutará *deathAnim*, mientras que, si dispara antes

de que aparezca la señal, se ejecutará *missedShot*. Cuando haya que empezar la siguiente ronda se disparará un *trigger* que hará que acaben estas animaciones y se pase directamente a *BackToIdle*, animación desde la que se repite el ciclo entero.



## 10. Conclusiones

---

Al respecto de los objetivos establecidos en el apartado 1.2, podemos afirmar que se han cumplido todos. Se han realizado dos MVPs funcionales con sus correspondientes experimentos, subiendo el videojuego a *Play Store* y promocionándolos por medio de redes sociales para llegar a conocidos y tener opiniones variadas. Además, se ha utilizado el *feedback* de la encuesta realizada en el primer MVP para mejorar el videojuego antes del segundo MVP y se utilizarán las respuestas de la segunda encuesta para conseguir futuras mejoras.

Además, se ha indagado en el mundo del emprendimiento y se han elaborado los documentos necesarios para analizar el mercado como por ejemplo el *Lean Canvas* o la Matriz DAFO. Se han utilizado metodologías ágiles y se ha organizado el trabajo por medio de la herramienta *Worki*, haciendo también reuniones diarias entre los miembros del equipo. Además, ya que debido a la pandemia de la COVID-19, las reuniones se hacían de forma *online*, se efectuaba mensualmente una reunión presencial para facilitar la comunicación entre los miembros del equipo.

Respecto al tercer objetivo, el juego ha sido publicado en la *Play Store* y es accesible a todo el mundo simplemente escribiendo *Blitz Party* en el buscador. El cuarto objetivo también ha sido cumplido, como se ha elaborado en el capítulo siete de este documento.

Cabe mencionar que a pesar de que, entre los cuatro miembros principales el trabajo ha funcionado de forma fluida y sin dificultades, ha habido ciertos conflictos con la persona encargada del arte, externa al equipo, que decidió abandonar el proyecto a mitad. Esto provocó que hubiera que buscar *assets* gratuitos y de uso libre para elaborar el contenido del videojuego.

Para acabar, destacamos el aprendizaje que ha supuesto la experiencia de desarrollar un videojuego online como proyecto de emprendimiento debido a que ha sido necesario familiarizarse con nuevas tecnologías para la parte técnica y además ha habido que elaborar muchos documentos con los que, a pesar de que se tenía experiencia debido a algunas asignaturas como Proyecto de Ingeniería de Software o Fundamentos de Organización de Empresas, ha sido necesario por primera vez usarlos en una situación real.

# 11. Trabajo futuro

---

El proyecto final ha resultado en un producto completamente terminado, con las funcionalidades básicas de un videojuego (ajustes, guardado de datos, tienda, inventario, etc.) y tres minijuegos que se pueden jugar tanto en multijugador en red local como en modo un jugador contra la IA. Partiendo de este punto, podemos dividir los próximos pasos a seguir en tres apartados distintos.

Primero, se espera aumentar el contenido disponible, mejorando la experiencia de juego y la diversión ofrecida mediante la implementación de más minijuegos que doten al producto de mayor diversidad a la hora de elegir. Los minijuegos elegidos se escogerán en base a lo diferentes que sean en comparación a los que ya están disponibles.

En segundo lugar, se prevé añadir nuevos objetos temáticos a la tienda, que aumenten el catálogo de accesorios que se pueden comprar y equipar a nuestro personaje. Así, por ejemplo, habrá prendas de ropa navideñas cuando se acerque la época navideña o accesorios de Pascua cuando sean estas fechas. Adicionalmente, se planifica monetizar el juego e introducir un sistema de adquisición de monedas de juego o *acorns* (también conocido como pase de batalla) que se pueda obtener por un módico precio y recompense a los jugadores con *acorns* adicionales.

Finalmente, se pretende mejorar el modo multijugador adquiriendo un servidor dedicado, realizando una modificación del sistema y pasando del multijugador local al online, con un *matchmaking* personalizado y la posibilidad de jugar contra personas de todo el mundo.



# 12. Apéndices

---

## 12.1 Listado de abreviaturas, siglas y acrónimos

TFG: Trabajo de Fin de Grado.

MVP: Minimum Viable Product (Producto Mínimo Viable, en español).

ETSINF: Escuela Técnica Superior de Ingeniería Informática.

DAFO: Debilidades, Amenazas, Fortalezas y Oportunidades.

IP: Intellectual Property (Propiedad Intelectual, en español).

GDD: Game Design Document (Documento de Diseño del Juego, en español).

MMO: Massively Multiplayer Online (Videojuego Multijugador Masivo en Línea, en español).

IA: Inteligencia Artificial.

2D: Dos Dimensiones.

3D: Tres Dimensiones.

IDE: Entorno de Desarrollo Integrado.

API: Application Programming Interface (Interfaz de Programación de Aplicaciones, en español).

SDK: Software Development Toolkit

## 12.2 Lista de recursos gráficos

Diseñados por los miembros del equipo de este TFG:

- Logo de Blitz Party.
- Título de Blitz Party, presente en la pantalla de registro.
- Menús del juego, incluyendo los botones e iconos que contienen.
- Animaciones del menú de ajustes, del menú multijugador y de la ventana de búsqueda de rival.
- Diseños del Zoomy, el personaje del jugador.
- Fondo de Binky Pursuit.

- Fondo de Whack-a-Mole.
- Zoomies de Whack-a-Mole.

Obtenidos de otras fuentes:

- Zorros y sus animaciones en Binky Pursuit y Cowboy Duel:  
<https://assetstore.unity.com/packages/2d/characters/sunny-land-103349>
- Binkies y sus animaciones en Binky Pursuit:  
<https://assetstore.unity.com/packages/2d/characters/sunny-land-103349>
- Red de Binky Pursuit: [https://favpng.com/png\\_view/net-the-legend-of-zelda-skyward-sword-animal-crossing-new-leaf-the-legend-of-zelda-ocarina-of-time-the-legend-of-zelda-a-link-to-the-past-png/KM8beq3G](https://favpng.com/png_view/net-the-legend-of-zelda-skyward-sword-animal-crossing-new-leaf-the-legend-of-zelda-ocarina-of-time-the-legend-of-zelda-a-link-to-the-past-png/KM8beq3G)
- Reloj de arena del cronómetro de los minijuegos: <https://www.pngegg.com/es/png-bbnuc>
- Topo de Whack-a-Mole: <https://www.pngwing.com/en/free-png-irbzl>
- Martillo de Whack-a-Mole: <https://www.nintenderos.com/2015/10/el-jefe-final-de-la-zona-caverna-protagoniza-la-captura-del-dia-de-zelda-tri-force-heroes/>
- Fondo de Cowboy Duel: <https://designbundles.net/onyxprj/766599-vector-cartoon-seamless-landscape-with-stones-and->
- Iconos de los objetos de la tienda: <https://www.cleanpng.com/>
- Fondo de la pantalla de carga: <https://www.bigstockphoto.com/es/image-162844712>



## 12.3 Game Design Document (GDD)



### *Game Design Document para:*



All work Copyright ©2021 by SwifterGames

Written by  
Jose Francisco Gómez Alemany  
Pablo Stemmer Fernández  
Alejandro Vicent Micó  
Marc Garcia Ferrer

Version #3.0

Domingo, 20 de Junio de 2021

1. Historial de diseño	99
Versión 1.00	99
Versión 2.00	99
Versión 3.00	99
2. Visión global del juego	99
Filosofía	99
Preguntas frecuentes	99
3. Conjunto de características	100
Características generales	100
Características multijugador	101
Editor	101
Jugabilidad	101
4. El mapa del mundo	101
Visión general	101
Zonas temáticas	101
Parque en constante evolución	102
Sistema de renderizado	102
Cámara	102
Motor de videojuegos	103
5. Personajes del juego	103
Visión general	103
Creando un personaje	104
Personajes especiales para cada minijuego	104
6. Interfaz de usuario	104
Visión general	104
7. Herramientas	107
Visión general	107
Variedad de herramientas	107
8. Banda sonora y efectos de sonido	108



Visión general	108
Sonido 3D	108
9. Minijuegos	108
Binky Pursuit	108
Whack-a-Mole	111
Cowboy Duel	114
10. Juego para un jugador	117
Visión general	117
Historia	117
11. Juego multijugador	117
Visión general	117
Batalla de minijuegos	118
Máximo de jugadores	118
Servidores	118
Personalización	118
Internet	118
Persistencia	118
Guardado y cargado	118

# 1. Historial de diseño

El objetivo del historial de diseño es que queden claros los cambios hechos en cada versión del GDD. Se irá actualizando según vaya cambiando este documento.

## Versión 1.00

La versión 1.00 es la primera versión de este documento. Se han añadido todas las secciones que están en el índice y se ha escrito su contenido.

## Versión 2.00

La versión 2.00 es la segunda versión de este documento. Se han actualizado las secciones anteriores y se ha añadido el apartado minijuegos con los datos de sus respectivos minijuegos.

## Versión 3.00

La versión 3.00 es la tercera versión de este documento. Se ha mejorado el formato del documento y se han actualizado diversos capítulos.

# 2. Visión global del juego

## Filosofía

### Diversión rápida

El principio más importante de este juego es la diversión rápida. Es decir, que no haya que dedicarle un gran número de horas al juego para que sea divertido, sino que se pueda coger en cualquier momento, jugar una partida y pasarlo bien.

### Espíritu competitivo

A pesar del estilo despreocupado que tiene el juego, queremos que los jugadores quieran volverse buenos en el juego y vencer tanto a sus amigos como a los rivales online, no solo por las recompensas sino también por la satisfacción de saber que han mejorado en el juego.

## Preguntas frecuentes

### ¿Qué es este juego?

*Blitz Party* es un juego online para móvil que consiste en batallas de minijuegos online. El juego encuentra a un rival y se juegan una serie de pequeños minijuegos en los que el objetivo es derrotar al rival de diferentes formas, ya sea siendo más rápido, más hábil o más listo.



### **¿Por qué crear este juego?**

Creamos este juego porque no consideramos que haya muchos juegos de este estilo en teléfonos móviles y creemos que es una plataforma a la que le sientan particularmente bien dado que es un género rápido y que no requiere nada para resultar divertido.

### **¿Dónde transcurre este juego?**

*Blitz Party* ocurre en el parque de *Rainbow Land*, un parque temático con gran multitud de zonas como por ejemplo una zona ambientada en el salvaje oeste. Los minijuegos de cada zona del parque contarán con su respectiva ambientación.

### **¿Qué controlo yo?**

El jugador controla a unos pequeños zorros que son la mascota de una especie de animales llamados *Zoomies*, con los cuales deberás superar los diferentes minijuegos de cada zona que encontrarás en el parque temático.

### **¿A cuántos personajes controlo?**

Cada jugador dispondrá de un personaje personalizable único que verá en el menú del juego. Sin embargo, en los minijuegos, controlará a la mascota de este, un pequeño zorro.

### **¿Cuál es el objetivo principal?**

Los *Zoomies* participan junto a sus mascotas en las batallas de minijuegos con el objetivo de convertirse en los futuros reyes de los juegos. Deberán desafiar a otras parejas de *Zoomies* con sus mascotas para ganar los máximos puntos posibles y alzarse con la victoria.

### **¿Qué lo distingue de otros juegos?**

Este formato de minijuegos no es tan común encontrarlo en dispositivos móviles.

## **3. Conjunto de características**

### **Características generales**

- Mapa con gran diversidad temática
- Gráficos 2D
- Personajes carismáticos y personalizables

## Características multijugador

- Partidas muy rápidas
- Posibilidad de jugar con tus amigos
- Sistema de ranking
- Recompensas por juego online
- Chat de emoticonos
- Inicio de sesión con cuenta propia y de terceros

## Editor

- Color del *Zoomy* personalizable
- Elementos cosméticos que añadir al *Zoomy*

## Jugabilidad

- Minijuegos frenéticos
- Aprovechamiento de las características del teléfono:
  - Pantalla táctil
  - Giroscopio
  - Sensor de movimiento
- Juego online con búsqueda automática de rival o amigos
- Juego en red local para jugar contra jugadores cercanos
- Posibilidad de juego *offline* para un jugador
- Guardado de récords y estadísticas tanto locales como globales

# 4. El mapa del mundo

## Visión general

El juego se desarrolla en el parque temático de *Rainbow Land*, donde existen diversas zonas temáticas: zona del salvaje oeste, zona acuática, zona futurista, zona volcánica, zona glacial, zona aérea, zona de prado, zona urbana, zona boscosa, zona egipcia, zona de los dragones.

## Zonas temáticas

Algunas de las zonas que tendrá el parque de *Rainbow Land* son:

- **Zona del salvaje oeste:** zona con minijuegos como duelo de vaqueros, carreras de caballos y disparo a las latas.
- **Zona acuática:** zona con minijuegos como pesca, remo, y búsqueda de tesoros en el fondo marino.



- **Zona futurista:** zona con minijuegos como pelea de boxeo con robots, cadena de montaje de robots y vuelo con jetpack.
- **Zona volcánica:** zona con minijuegos como salto de río de lava, esquivar bolas de fuego y escapar de una erupción.
- **Zona glacial:** zona con minijuegos como hockey sobre hielo, carrera de patinaje y curling.
- **Zona aérea:** zona con minijuegos como paracaidismo, vuelo con obstáculos y piruetas con avioneta.
- **Zona de prado:** zona con minijuegos como recolección de setas, persecución de conejos y aplastar al topo.
- **Zona urbana:** zona con minijuegos como cocinar, trabajar en la oficina y limpiar las ventanas.
- **Zona boscosa:** zona con minijuegos como capturar hadas, encontrar al duende y talar árboles.
- **Zona egipcia:** zona con minijuegos como escapar de la pirámide, enrollar a la momia y surfear en las dunas.
- **Zona de los dragones:** zona con minijuegos como lanzar shurikens a la diana, pelea de samurais y derrotar al dragón.

## Parque en constante evolución

El parque no está completamente terminado. Se irán añadiendo nuevas zonas con el paso del tiempo y las antiguas zonas obtendrán nuevos minijuegos para mantener la frescura de la jugabilidad.

## Sistema de renderizado

### Visión General

El juego se desarrollará en el motor de videojuegos *Unity*, y este incorpora un sistema integrado para renderizar tanto en 2D como 3D. Para ello, puede hacer uso de su anterior *Built-In Renderer* como del nuevo *Universal Render Pipeline* (URP) optimizado tanto para gráficos 2D como 3D.

### Renderizado 2D/3D

Para el caso de este juego 2D, se usará la configuración de *Unity* especializada en la renderización de assets 2D y para manipular y trabajar cómodamente con ellos.

## Cámara

### Visión General

El ángulo y comportamiento de la cámara dependerá del minijuego que se esté jugando en ese momento.

### **Cámara dependiente del minijuego**

En algunos minijuegos la cámara será *top-down*, por ejemplo, en minijuegos de carreras, para poder ver mejor los obstáculos y el rival. En otros, como el de surf de dunas o vuelo con obstáculos habrá *scroll* lateral para así dar sensación de avance y en otros, como el de esquivar bolas de fuego o captura de hadas la cámara será fija para dar mejor control sobre el movimiento.

### **Motor de videojuegos**

#### **Visión General**

El motor de videojuegos usado es *Unity*. *Unity* es un motor multiplataforma que permite un desarrollo cómodo y completo de videojuegos tanto en 2D como en 3D, ofreciendo a los desarrolladores herramientas especializadas en los diferentes campos del desarrollo. Para el desarrollo se usarán las herramientas dedicadas al 2D como puede ser el motor de físicas Box2D, iluminación 2D, tilemap, colisiones 2D, herramientas para trabajar con sprites, etc.

## **5. Personajes del juego**

### **Visión general**

Nuestros personajes son seres de la especie de los *Zoomies*. Estos seres presentan características similares a los de una ardilla, pero un cuerpo más redondo. Los *Zoomies* son unos seres muy competitivos y les encanta el sabor de la victoria y por ello disfrutan participando en los minijuegos que el señor Theodore ha preparado para ellos. Los personajes que controlamos dentro de los minijuegos son las mascotas y compañeros de nuestros *Zoomies*. Al poco de nacer los *Zoomies* tienen la posibilidad de elegir a su mascota, un pequeño zorro recién nacido. Al ser elegido, se crea un vínculo inquebrantable entre ellos que los une para siempre.

El señor Theodore es el *Zoomy* detrás del parque *Rainbow Land* y es un amante de la diversión y la adrenalina. Se le ocurre crear el parque para que los demás *Zoomies* se diviertan como hacía él en sus años mozos. Además, cada zona cuenta con un *Zoomy* especial que son los encargados de dicha zona. En la Figura 101 podemos ver los bocetos originales de los *Zoomies* junto a un simple modelo.





Figura 101. Personajes de *Blitz Party*

## Creando un personaje

Al empezar el juego el jugador deberá elegir el color de su *Zoomy*. Este color más adelante podrá ser cambiado cuando el jugador desbloquee nuevos colores usando las recompensas que obtienen jugando y podrán añadirle accesorios cosméticos para personalizarlos a su gusto. Habrá un abanico de elementos cosméticos como sombreros, peinados, ropa etc.

## Personajes especiales para cada minijuego

En algunos minijuegos, como en el de captura de hadas o el de vuelo con *jetpack*, aparecerán personajes especiales que tendrán impacto solo en ese minijuego. Por ejemplo, las hadas aparecerán en su minijuego y el objetivo será capturarlas. En el de vuelo con *jetpack*, aparecerán pájaros que el jugador tendrá que esquivar para no perder tiempo y altura.

# 6. Interfaz de usuario

## Visión general

Cuando se abre el juego entramos a la interfaz principal del juego. Desde esta interfaz podemos navegar a otras pestañas:

- Ajustes: Presionando sobre el engranaje
- Tienda: Presionando sobre el botón con el signo del dólar
- Customizar: Presionando sobre el botón la foto de un *Zoomy*
- Amigos: Presionando sobre el botón con las siluetas humanas
- Perfil: Presionando sobre el botón con la imagen de un carné

Al presionar ajustes se abrirá una ventana por encima donde el usuario podrá cambiar diversos ajustes además de entrar a los créditos y habrá un botón para salir del juego. Para cerrar esta pestaña y volver a la interfaz principal simplemente se deberá presionar la X en la esquina superior derecha.

En la ventana de customización se verá al *Zoomy* actual del jugador y tendrá diversas pestañas, cada pestaña representando distintos tipos de elementos de personalización como peinados, colores, trajes completos etc. Al tocar uno de estos elementos el *Zoomy* del menú principal verá reflejado el cambio. Es decir, si el jugador toca un peinado, el *Zoomy* obtendrá ese peinado para ver como quedaría y decidir si le gusta y mantenerlo o elegir otro peinado.

En la tienda el jugador podrá ver los elementos que hay disponibles para comprar a cambio de *acorns* (bellotas), la moneda del juego. Los elementos disponibles para comprar en la tienda cambiarán cada X horas. Además de poder comprar elementos de personalización, los jugadores podrán comprar la moneda premium.

En la ventana de amigos, el jugador podrá añadir nuevos amigos usando un sistema de códigos de amigo exclusivo a cada jugador. Desde esta ventana además se pueden desafiar a una batalla de minijuegos a los amigos que ya tengamos agregados.

En la ventana de perfil de usuario podemos iniciar sesión (si se está usando una cuenta de invitado), ver el nombre de usuario y el código de amigo y se puede acceder al historial. Por otra parte, en la Figura 102 se observa el diseño inicial del menú principal del videojuego.





Figura 102. Mockup menú principal *Blitz Party*

### Leyenda:

- A: Ajustes
- I: Pantalla Inicial
- C: Personalización de *Zoomy*
- T: Tienda
- Am: Amigos
- P: Perfil de usuario

Cuando se presiona el botón jugar se accede a una pantalla donde se puede elegir entre modo de juego libre o modo batalla (Figura 103). En el caso de juego libre, el jugador elegirá un minijuego para jugar de manera *offline* contra una IA. En el modo batalla, se podrá elegir entre jugar una batalla en red local, contra jugadores físicamente cercanos, o en modo online para jugar contra jugadores aleatorios de todo el mundo. El modo batalla es una serie de tres minijuegos completamente aleatorios. El jugador que se alce con más victorias será nombrado vencedor.

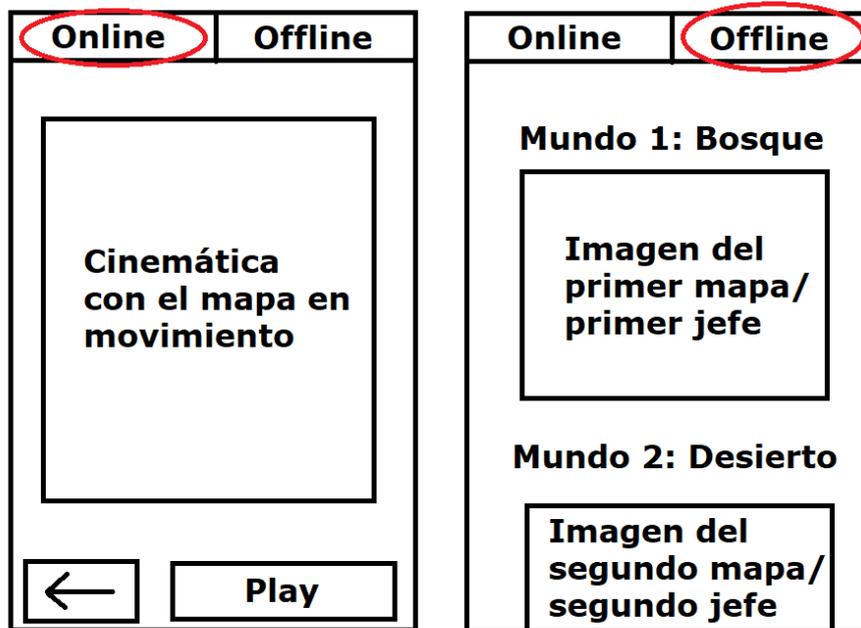


Figura 103. Pantallas de juego

## 7. Herramientas

### Visión general

En algunos minijuegos, los *Zoomies* tendrán herramientas especiales únicamente para el desarrollo de ese minijuego.

### Variedad de herramientas

Habrán muchos tipos de herramientas dependiendo del minijuego. Por ejemplo: en el minijuego de duelo de vaqueros, el personaje llevará un arma de fuego. En el de volar con *jetpack*, obviamente llevará un *jetpack* y en el de pescar utilizará una caña. En la Figura 104 se pueden observar ejemplos de herramientas que se utilizarán en estos minijuegos.



Figura 104. Herramientas de los minijuegos

## 8. Banda sonora y efectos de sonido

### Visión general

La banda sonora del juego consistirá en piezas musicales acordes a las circunstancias en las que se encuentra el juego. La música de fondo es una pieza alegre que incita la exploración de las diferentes zonas del parque temático, mientras que cada minijuego tendrá una banda sonora más frenética inspirada en la temática de dicho minijuego.

### Sonido 3D

Para la implementación de la banda sonora y todo tipo de clips sonoros, se hace uso de las herramientas incluidas en el motor *Unity*. Éste consta de componentes tales como *Audio Listener*, *Audio Clip*, *Audio Source*, etc, para la fácil manipulación de sonidos haciendo uso de *Audio Mixers*.

## 9. Minijuegos

### Binky Pursuit

#### Descripción

Minijuego en el que dos jugadores se batan para ver quién es capaz de capturar más *Binkies*. En modo de un solo jugador, el usuario se bate contra una IA mientras que en multijugador se enfrenta a otro jugador.

#### Historia

Se ha encontrado una plaga de *Binkies* en el claro del bosque. Que gran oportunidad para realizar un pequeño duelo. ¡A ver quién captura más *Binkies*!

#### Controles

Para controlar al personaje el jugador debe tocar el lado izquierdo de su dispositivo móvil para que aparezca el joystick y a partir de aquí deslizar hacia cualquier dirección para que el movimiento se refleje en el personaje.

Para capturar a los *Binkies*, los jugadores deben colocarse cerca de ellos y presionar el botón de captura que está en el lado derecho de la pantalla. Si el jugador toca a algún *Binky* este desaparecerá y se le sumará un punto.

## Escenario

El escenario (Figura 105) simula el claro de un bosque donde hay cuatro madrigueras dispuestas de manera que a cada costado de la pantalla quede una. En los bordes del mapa existen filas de árboles que marcan los límites de la zona jugable.

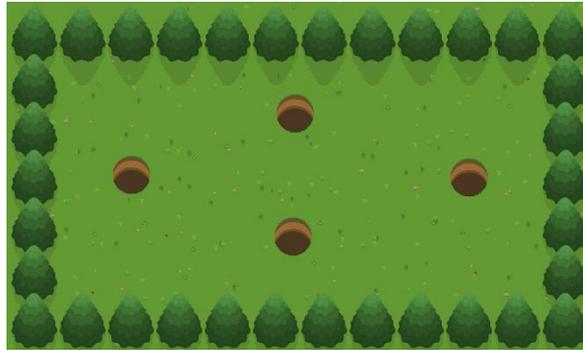


Figura 105. Escenario de *Binky Pursuit*

## Interfaz de usuario

Como se aprecia en la Figura 106, en la esquina superior izquierda de la pantalla podemos observar la puntuación del jugador 1 junto a un pequeño avatar del personaje. A su derecha está el contador que marca el tiempo restante. Después está la puntuación del jugador dos y finalmente, tenemos el botón de pausa que es el responsable de pausar la partida y desde donde podemos volver a la partida o al menú principal.

En la parte inferior de la pantalla, en el lado izquierdo se encuentra el joystick el cual mueve al personaje del jugador. Sin embargo, este joystick tan solo aparece si existe contacto con la pantalla en la mitad izquierda de la pantalla. En el lado derecho vemos el botón de captura usado para capturar a los *Binkies* cuando el personaje está encima.

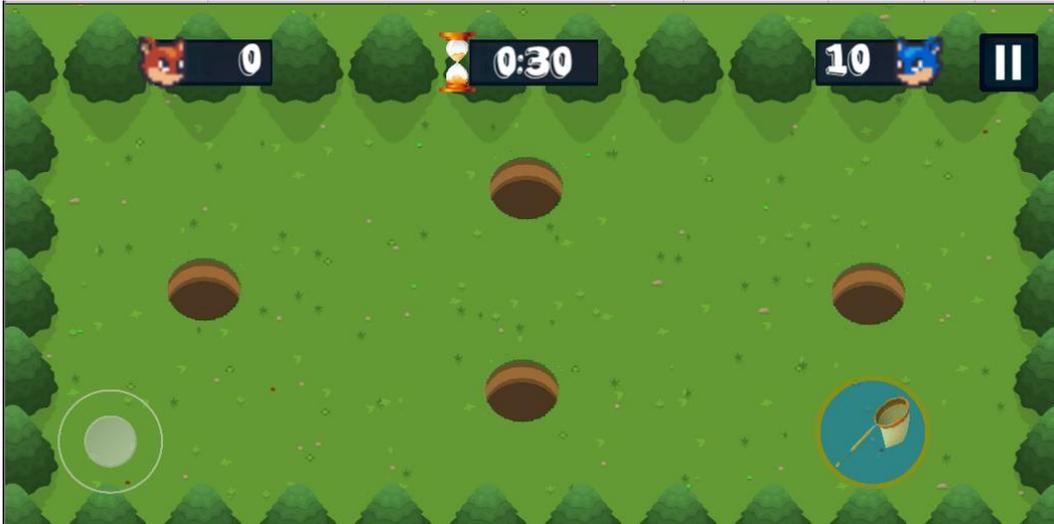


Figura 106. Interfaz de *Binky Pursuit*

### Personajes

Existen 2 tipos de personajes en este minijuego: los zorros y los *Binkies*.

- Zorros (Figura 107): Los personajes controlables por los jugadores.



Figura 107. Zorros de *Binky Pursuit*

- *Binkies* (Figura 108): Los enemigos del minijuego. Salen de sus madrigueras y se mueven de manera aleatoria. En el caso de que entren a una madriguera desaparecen y los jugadores pierden la oportunidad de capturarlo.



Figura 108. *Binkies* de *Binky Pursuit*

### Efectos visuales y sonoros

Este minijuego tiene una canción de fondo vibrante y bastante animada que suena durante todo el minijuego. Hay un sonido cuando el jugador captura un *Binky* y suma un punto. Cuando el tiempo acaba y sale el panel de resultados, suena un sonido de victoria, derrota, o empate dependiendo del resultado.

Los zorros tienen dos animaciones una en estado *idle* que está activa cuando están quietos y otra de correr que se activa cuando se mueven. Los *Binkies* tienen una sola animación ya que están constantemente en movimiento.

### **Dinámica del juego**

Al entrar al minijuego cada jugador aparece en un lado y empieza la cuenta atrás. A partir de este momento comenzarán a salir *Binkies* de las madrigueras y los jugadores deben perseguirlos y atraparlos. Cuando un jugador atrape a un *Binky*, este desaparecerá y se le otorgará un punto al jugador que lo haya atrapado. Conforme pasa el tiempo, el ritmo al que aparecen los *Binkies* va aumentando e incluso pueden salir hasta cuatro de golpe. Al acabar el tiempo aparece un panel con el resultado de la partida y las respectivas puntuaciones de los jugadores.

### **Condición de victoria**

Resultará vencedor el jugador que tenga la mayor puntuación al finalizar el tiempo, es decir, el jugador que más *Binkies* haya atrapado.

## **Whack-a-Mole**

### **Descripción**

Minijuego en el que dos jugadores se batan para ver quien alcanza una puntuación más alta golpeando a los topos que aparecen de los distintos agujeros. En modo de un solo jugador, el usuario debe intentar superar la puntuación establecida la cual simula la puntuación del segundo jugador. En modo multijugador se enfrentan directamente dos jugadores.

### **Historia**

Los topos están desatados. ¡Golpéales con el martillo cuando salgan de sus madrigueras, pero cuidado con los *Zoomies*!

### **Controles**

En este minijuego un simple toque de pantalla ejecuta la animación de golpeo del martillo donde el jugador ha tocado.

### **Escenario**

El escenario (Figura 109) simula una llanura donde hay siete madrigueras de topos. En los bordes del mapa existen vallas que delimitan el mapa.



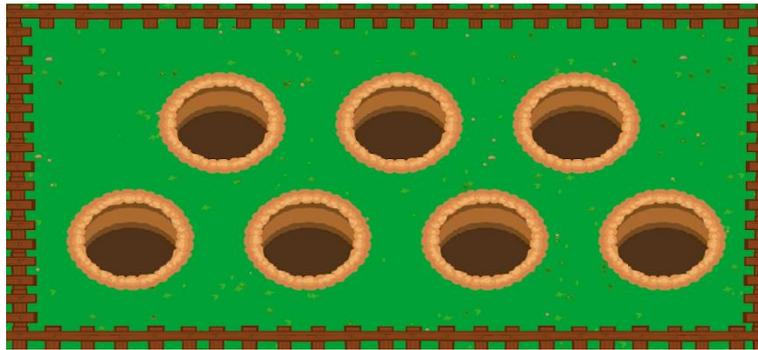


Figura 109. Escenario de *Whack-A-Mole*

### Interfaz de usuario

Como se puede ver en la Figura 110, en este minijuego volvemos a tener los componentes básicos formado por las dos puntuaciones, el contador de tiempo y el botón de pause. Debido a los controles de este minijuego, no es necesario ningún tipo de componente visual adicional.



Figura 110. Interfaz de *Whack-A-Mole*

### Personajes

Existen cuatro tipos de personajes en este minijuego: el martillo, los topos, los topos dorados y los *Zoomies*.

- Martillo (Figura 111): es el personaje que controla el jugador. Simula el martillo real del arcade original Whack-A-Mole (Guacamole en España).



Figura 111. Martillo de *Whack-A-Mole*

- Topo (Figura 112): es el enemigo estándar. Golpearle con el martillo otorga un punto.



Figura 112. Topo normal de *Whack-A-Mole*

- Topo dorado (Figura 113): es un topo especial que tiene menor probabilidad de aparecer que el topo. Golpearle con el martillo otorga cinco puntos.



Figura 113. Topo dorado de *Whack-A-Mole*

- Zoomy (Figura 114): es el personaje del jugador. Golpearle con el martillo resta tres puntos.



Figura 114. *Zoomy* de *Whack-A-Mole*

### **Efectos visuales y sonoros**

Este minijuego tiene una canción de fondo con un tema frenético que suena durante todo el minijuego. El martillo realiza un sonido al no golpear a ningún personaje. Cada personaje golpeable (topo, topo dorado y *Zoomy*) realiza un sonido al ser golpeado. Cuando el tiempo acaba y sale el panel de resultados, suena un sonido de victoria, derrota, o empate dependiendo del resultado.

El martillo realiza una animación de golpeo cuando el jugador toca la pantalla. Los otros tres personajes tienen tres animaciones:

1. Salir de la madriguera.
2. Volver a entrar a la madriguera.
3. Ser golpeados

### **Dinámica del juego**

Al entrar al minijuego el jugador toma el control del martillo y empieza la cuenta atrás. A partir de este momento comenzarán a salir personajes de las madrigueras, ya sean topos, topos dorados y *Zoomies*, y los jugadores deben tocarlos antes de que vuelvan a entrar a las madrigueras. Cuando un jugador golpee a un personaje, este realizará una pequeña animación y desaparecerá y los puntos se actualizarán dependiendo del personaje golpeado. Conforme pase el tiempo, el ritmo a la que aparecen los personajes de las madrigueras va aumentando hasta que llega a una velocidad extremadamente rápida. Al acabar el tiempo aparece un panel con el resultado de la partida y las respectivas puntuaciones de los jugadores.

### **Condición de victoria**

Resultará vencedor el jugador que tenga la mayor puntuación al finalizar el tiempo.

## **Cowboy Duel**

### **Descripción**

Minijuego en el que dos jugadores se batan en un clásico duelo del lejano oeste. El primer jugador que dispare a partir de la señal de disparo será declarado vencedor. En modo de un solo jugador, el usuario deberá vencer a la IA mientras que en modo multijugador se batirá en duelo contra otro jugador.

### **Historia**

Un duelo como los de antes. ¡Espera hasta la señal y dispara antes que tu rival!

### **Controles**

En este minijuego un simple toque de pantalla ejecuta el disparo.

### **Escenario**

El escenario (Figura 115) simula un desierto del lejano oeste donde los vaqueros solían tener este tipo de duelos.



Figura 115. Escenario de *Cowboy Duel*

### Interfaz de usuario

Como se puede apreciar en la Figura 116, la UI de este minijuego está formada por la puntuación de ambos personajes, el botón de pausa y en este caso una pequeña cuenta atrás formada por 3, 2, 1, *Shoot*. Entre los números pasa un segundo de tiempo y el texto de Shoot, que es la marca que permite a los jugadores disparar, aparece aleatoriamente después para que los jugadores no puedan controlar los tiempos.



Figura 116. Interfaz de *Cowboy Duel*

### Personajes

En este minijuego participan los dos zorros, uno de cada jugador (Figura 117).



Figura 117. Personajes de *Cowboy Duel*

### **Efectos visuales y sonoros**

Este minijuego tiene una canción de fondo que produce tensión y que recuerda a un duelo del oeste. Durante la cuenta atrás suena un sonido cada segundo y otro que da la señal que marca que los jugadores deben disparar. Al disparar suena un sonido de disparar y el de una bala cayendo. Cuando el tiempo acaba y sale el panel de resultados, suena un sonido de victoria o derrota.

Al comenzar cada ronda los personajes realizan una pequeña animación en la que se preparan para el duelo moviéndose cada uno a un lado de la pantalla. Disparar también realiza una animación en la que sale el arma y se dispara la bala. Además, existe una animación de ‘muerte’ para el jugador que pierde la ronda.

### **Dinámica del juego**

Al entrar al minijuego veremos una pequeña escena donde los dos zorros que comienzan cara a cara, se giran y andan ciertos pasos en direcciones opuestas. En este momento el jugador toma el control y ya puede disparar. Saldrá una pequeña cuenta atrás y en un momento aleatorio aparecerá la señal de disparar. El primer jugador que presione la pantalla para disparar a partir de este punto ganará la ronda. En caso de que se dispare antes de la señal el jugador perderá la bala y no podrá disparar con lo cual perderá la ronda. En caso de que ningún jugador dispare se realizará un disparo forzado. Se jugará hasta un máximo de tres rondas.

### **Condición de victoria**

Resultará vencedor el jugador que obtenga dos victorias, es decir, al mejor de tres.

## 10. Juego para un jugador

### Visión general

En el juego offline podrás jugar al modo libre donde podrás jugar a los minijuegos con total libertad.

### Historia

Tanto los *Zoomies* como sus compañeros son unos seres extremadamente competitivos y no hay nada que les guste más que una buena victoria. Existe una competición de gran prestigio llamada los *ZoomyJuegos*. Esta competición es una ardua batalla de minijuegos donde se ponen a prueba múltiples habilidades, tanto físicas como mentales. El sueño de todo *Zoomy* es ganar esta competición y alzarse con el gran premio: la bellota dorada.

Mr Theodore, un *Zoomy* ya de edad avanzada, fue el ganador de los *ZoomyJuegos* durante diez años seguidos y es una leyenda entre los *Zoomies*. Todo *Zoomy* aspira a llegar a ser alguien como él y convertirse en el mejor minijugador de todos los tiempos.

Mr Theodore, como amante de la diversión y de la adrenalina, crea *Rainbow Land*, un parque temático donde los *Zoomies* y sus compañeros pueden divertirse y entrenar para participar en los siguientes *ZoomyJuegos*. El jugador es uno de estos *Zoomies* que entrenan para tener una posibilidad de llevarse la victoria en los *ZoomyJuegos*.

El parque está separado por zonas y cada zona está representada por uno de los antiguos rivales de Mr Theodore y serán los mayores desafíos a los que se enfrentarán los *Zoomies*.

## 11. Juego multijugador

### Visión general

El usuario dispondrá de la posibilidad de elegir entre jugar solo o en modo multijugador. En caso de buscar partida online, el jugador será emparejado aleatoriamente con otro jugador de cualquier parte del mundo. Ambos se batirán en un duelo en el que deberán superar una serie de minijuegos aleatorios del total disponible basados en las diferentes zonas del parque temático.



## **Batalla de minijuegos**

La batalla de minijuegos es una serie de tres minijuegos aleatorios en los que el jugador se enfrentará a un jugador cercano si escoge la batalla en red local o a un jugador aleatorio en caso de jugar online. El jugador podrá seleccionar este modo para divertirse y competir sin tener que decidir a qué minijuegos jugar y a la vez poder ganar bellotas para poder comprarle complementos a su *Zoomy*.

## **Máximo de jugadores**

Todos los duelos multijugador constan de dos jugadores.

## **Servidores**

El juego seguirá una arquitectura cliente-servidor, donde los clientes, en este caso los jugadores, envían peticiones al servidor para ejecutar acciones y el servidor será el encargado de devolver el resultado y actualizar las diferentes estancias de juego de los clientes.

De esta forma se puede mitigar algunos sistemas de cheating por parte de los usuarios, ya que solo el servidor puede realizar las acciones y se realiza cierta autenticación para ejecutar código.

## **Personalización**

Los jugadores podrán crear duelos personalizados entre amigos agregados dentro del juego.

## **Internet**

Cuando un jugador le da a buscar partida multijugador, el juego le emparejará con otra persona que se encuentre en ese mismo instante buscando partida. Una vez finalizada la partida, se registrarán las puntuaciones de los usuarios.

## **Persistencia**

Para el modo multijugador solo necesita ser persistido todo tipo de dato relacionado con la cuenta del usuario como puede ser su sesión iniciada mediante un servicio de terceros, nombre de usuario, puntuación adquirida y su ranking global.

## **Guardado y cargado**

No se puede guardar o cargar en el modo multijugador porque esa funcionalidad no va a ser requerida debido al sistema de batallas que sigue el juego. Parejas de dos jugadores se batan en una serie de minijuegos y al finalizar la partida se almacenan las puntuaciones, pero no se precisa guardar o cargar ningún estado de avance en el juego como tal.

## 12. Calendario de trabajo

Tareas para completar y Calendario				
Tareas	Líder de Tarea	Inicio	Fin	% Completado
<b>Fase de Desarrollo</b>				
Diseño		15/02/21	06/07/21	
Historia	Marc	15/02/21	20/03/21	100%
Personajes	Alejandro	15/02/21	22/03/21	100%
Minijuegos	Jose Francisco	17/02/21	10/05/21	100%
Zonas	Pablo	17/02/21	05/04/21	100%
Mecánicas gameplay	Jose Francisco	20/03/21	06/07/21	100%
Arte		15/03/21	06/07/21	
Personajes	Alejandro	15/03/21	01/07/21	100%
Special FX	Jose Francisco	10/04/21	06/07/21	100%
UI	Alejandro	25/03/21	20/05/21	100%
Fondos	Marc	20/04/21	07/05/21	100%
Animaciones	Pablo	20/04/21	06/07/21	100%
Desarrollo		25/03/21	06/07/21	
Batallas online	Pablo	25/04/21	06/07/21	50%
Batallas en red local	Pablo	25/04/21	06/07/21	100%
Emparejamiento online	Pablo	20/04/21	06/07/21	50%
Juego offline	Jose Francisco	25/03/21	07/05/21	100%
IA enemigos	Marc	23/03/21	18/04/21	100%
Pathfinding		23/03/21	01/04/21	100%
Algoritmo Spawners		23/03/21	18/04/21	100%
Minijuegos	Marc	15/03/21	01/07/21	80%
Binky Pursuit		15/03/21	01/04/21	100%
Whack-A-Mole		02/04/21	18/04/21	100%
Cowboy Duel		19/04/21	10/05/21	100%
Horse Race		11/05/21	25/07/21	20%
Ice Hockey		26/05/21	10/06/21	0%
Personalización	Jose Francisco	01/05/21	01/06/21	50%
Partidas privadas	Pablo	10/06/21	06/07/21	0%
Audio				
Música	Jose Francisco	30/03/21	15/04/21	100%
Efectos de sonido	Marc	15/04/21	15/05/21	100%
Milestone: Características Principales		15/02/21	28/04/21	100%
Milestone: MVP 1		15/02/21	24/05/21	100%
Milestone: MVP 2		15/02/21	26/06/21	100%
<b>Fase de Testeo</b>				
Plan de pruebas	Jose Francisco	06/07/21	10/07/21	0%
Beta Testing	Alejandro	10/07/21	20/07/21	0%
Milestone: Pruebas QA		06/07/20	20/07/21	0%
<b>Fase de Despliegue</b>				
Listo para salida	Pablo	20/07/21	30/07/21	0%
Milestone: Listo para uso		20/07/21	30/07/21	0%

Figura 118. Calendario de trabajo





## 13. Referencias bibliográficas

---

- [1] VentureBeat: *SuperData: Games hit \$120.1 billion in 2019, with Fortnite topping \$1.8 billion.* <<https://venturebeat.com/2020/01/02/superdata-games-hit-120-1-billion-in-2019-with-fortnite-topping-1-8-billion/>>
- [2] BusinessofApps: *Among Us Revenue and Usage Statistics (2021).* <<https://www.businessofapps.com/data/among-us-statistics/>>
- [3] Video Game Sales Wiki: *Mario Party* <[https://vgsales.fandom.com/wiki/Mario\\_Party](https://vgsales.fandom.com/wiki/Mario_Party)>
- [4] Video Game Sales Wiki: *Wario* <<https://vgsales.fandom.com/wiki/Wario>>
- [5] Adage: *How 'Dumb Ways to Die won the Internet, became the no. 1 campaign of the year.* <<https://adage.com/article/special-report-the-awards-report/dumb-ways-die-dissected/245195>>
- [6] PlayFab Community: Password Security <<https://community.playfab.com/questions/892/210521827-Password-security-.html>>

