



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 *front-end*

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor:** Alejandro Vicent Micó

**Tutor:** Francisco José Abad Cerdá

2020-2021



# Resumen

---

Este proyecto tiene como objetivo la implementación de un videojuego en el entorno de desarrollo *Unity*. El videojuego se ha realizado con un enfoque de emprendimiento dentro del marco de *Start.inf*, con proyección 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, 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.

La implementación del *front-end* es la parte principal de este trabajo, centrado en la programación de la lógica visible por el usuario. De esta forma, se muestra detalladamente los procesos desarrollados que interpretan las acciones del usuario y ayudan a abstraer su interacción en la capa de presentación, separándola de las capas de lógica de negocio y persistencia. Para ello, se ha prestado especial atención al diseño de la interfaz gráfica, siguiendo una serie de principios que ayudan al usuario a entender el funcionamiento de la interfaz independientemente de cómo se haya implementado internamente.

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

# Resum

---

Aquest projecte té com a objectiu l'implementació d'un videojoc en l'entorn de desenvolupament *Unity*. El videojoc s'ha realitzat amb un enfocament d'emprenedoria dins del marc de *Start.inf*, amb projecció 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, amb títol *Blitz Party*, és un videojoc multijugador per a dispositius mòbils que perteneix al gènere dels jocs basats en minijocs. Es tracta d'una aplicació *free-to-play* que compta amb contingut adicional 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 un àrea del projecte.

L'implementació del *front-end* és la part principal d'aquest treball, centrat en la programació de la lògica visible per l'usuari. Així, es mostra detalladament els processos desenvolupats que interpreten les accions de l'usuari i ajuden a abstraure la seua interacció en la capa de presentació, separant-la de les capes de lògica de negoci i persistència. Per això, s'ha posat especial atenció al diseny de la interfície gràfica, seguint una serie de principis que ajuden l'usuari a entendre el funcionament de l'interfície independentment de com s'haja implementat internament.

**Paraules clau:** videojoc, Unity, minijocs, empenedoria, *front-end*.

## Abstract

---

This project aims to create a videogame using the Unity Game Engine. The video game has been made with an entrepreneurial approach within the framework of Start.inf, 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, 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. The 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 àrea of the project.

The main part of this work is the implementation of the front-end, focused on programming the logic that the user interacts with. This way, we show in detail the developed processes that interpret user actions and help to abstract user interaction in the presentation layer, separating it from the business logic and persistence layers. This has been done by paying special attention to the design of the graphic interface, following a list of principles that help the user to understand the behaviour of the interface independently of its internal implementation.

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

# Tabla de contenidos

---

1. Introducción.....	13
1.1 Motivación .....	14
1.2 Justificación del tema .....	15
1.3 Objetivos .....	15
1.4 Estructura de la memoria.....	16
2. Evaluación de la idea de negocio .....	17
2.1 Estado del arte .....	17
2.2 Estudio de mercado .....	20
2.2.1 Super Mario Party .....	21
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 .....	35
2.6 Conclusiones .....	38
3. Desarrollo de la idea de negocio.....	39
3.1 Mapa de características .....	40
3.2 Primer <i>Minimum Viable Product</i> .....	41
3.2.1 Desarrollo del <i>MVP</i> .....	42
3.2.2 Experimento.....	44
3.3 Segundo <i>Minimum Viable Product</i> .....	50
3.3.1 Desarrollo del <i>MVP</i> .....	51
3.3.2 Experimento.....	53
3.4 Despliegue del producto y <i>marketing</i> .....	55
4. Tecnologías utilizadas .....	57
4.1 Unity Engine .....	57



4.2 Visual Studio.....	57
4.3 Git/GitHub .....	58
4.4 Worki Process .....	59
4.5 Photoshop.....	59
4.6 Mirror.....	60
4.7 Azure PlayFab.....	60
5. Desarrollo del <i>front-end</i> .....	62
5.1 Introducción .....	62
5.2 Metodología .....	63
6. Análisis de requisitos.....	67
6.1 Ingeniería de requisitos .....	67
6.2 Elicitación de requisitos .....	67
6.2.1 <i>Brainstorming</i> .....	68
6.2.2 Entrevistas.....	69
6.3 Especificación de requisitos .....	70
7. Diseño de la solución.....	72
7.1 <i>Mock-ups</i> .....	72
7.2 Principios del diseño de interfaces .....	74
7.3 Diagrama de flujo.....	77
8. Implementación del producto .....	79
8.1 Menús del juego .....	79
8.2 Funcionamiento de la tienda y el inventario .....	86
8.2.1 <i>Script ShopManager</i> .....	86
8.2.2 <i>Script InventoryManager</i> .....	87
8.3 Refactoring.....	88
9. Resultados obtenidos y pruebas.....	91
9.1 Validación de requisitos .....	91
9.2 Encuesta primer MVP .....	92
9.3 Encuesta segundo MVP.....	93

10. Conclusiones.....	96
11. Trabajo futuro.....	98
12. Apéndices .....	99
12.1 Listado de abreviaturas, siglas y acrónimos.....	99
12.2 Lista de recursos gráficos .....	100
12.3 <i>Game Design Document</i> (GDD).....	101
13. Referencias bibliográficas .....	125



## Listado de figuras

---

Figura 1. Logo de Blitz Party .....	13
Figura 2. Enlace QR de descarga de Blitz Party .....	14
Figura 3. Lanzamiento de jabalina en Olympic Decathlon .....	17
Figura 4. Partida de Bomberman Party Edition .....	18
Figura 5. Personaje personalizable de la Nintendo Wii .....	19
Figura 6. Captura del videojuego Buzz!: el Gran Reto .....	19
Figura 7. Cronología juegos party o de minijuegos .....	20
Figura 8. Tabla comparativa de productos competidores. ....	21
Figura 9. Portada de Super Mario Party .....	21
Figura 10. Modo tablero de Super Mario Party .....	22
Figura 11. Minijuego de Super Mario Party .....	23
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 .....	25
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 .....	27
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 .....	29
Figura 22. Proyección económica año 1 .....	31
Figura 23. Proyección económica año 2 .....	32
Figura 24. Personal contratado .....	32
Figura 25. Proyección económica años 3 y 4 .....	33
Figura 26. Número de descargas año 5 .....	34
Figura 27. Resultado acumulado de ingresos .....	34
Figura 28. Lean Canvas de Blitz Party .....	37
Figura 29. Pitch Doc .....	40
Figura 30. Mapa de características de Blitz Party .....	41
Figura 31. Mapa de características del primer MVP .....	42
Figura 32. Minijuegos seleccionados del primer MVP .....	42
Figura 33. Captura del minijuego Binky Pursuit .....	43
Figura 34. Captura del minijuego Whack-A-Mole .....	43

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



Figura 72. Evolución del personaje .....	69
Figura 73. Columnas de especificar requisitos y diseñar .....	70
Figura 74. Mock-up menú principal .....	72
Figura 75. Mock-up selección de partida .....	73
Figura 76. Worki: Mock-up menú principal .....	74
Figura 77. Consistencia en el esquema cromático .....	75
Figura 78. Menús emergentes .....	75
Figura 79. Botones con iconos .....	76
Figura 80. Leyes de Gestalt .....	77
Figura 81. Diagrama de flujo .....	78
Figura 82. Ventana de registro .....	79
Figura 83. Ventana central .....	80
Figura 84. Ventana de la tienda .....	80
Figura 85. Ventana del inventario .....	81
Figura 86. Ventana de perfil de usuario.....	81
Figura 87. Menú selección de minijuego.....	82
Figura 88. Menú modo batalla .....	82
Figura 89. Menú de ajustes.....	83
Figura 90. Ventana de créditos .....	83
Figura 91. Pantalla de carga .....	84
Figura 92. Minijuego Binky Pursuit .....	84
Figura 93. Minijuego Whack-A-Mole.....	85
Figura 94. Minijuego Cowboy Duel.....	85
Figura 95. Menú de pausa .....	86
Figura 96. Script ShopManager.....	87
Figura 97. Funcionamiento de la lista deslizable .....	88
Figura 98. PurchaseItemHandler con refactoring .....	89
Figura 99. Pruebas de aceptación .....	91
Figura 100. Ejecución de una PA .....	92
Figura 101. PA ejecutadas satisfactoriamente .....	92
Figura 102. Pregunta MVP 2 interfaz intuitiva.....	93
Figura 103. Pregunta MVP 2 interfaz atractiva .....	94
Figura 104. Feedback de la interfaz gráfica.....	94
Figura 105. Pregunta MVP 2 feedback de la interfaz .....	95
Figura 106. Personajes de Blitz Party .....	109
Figura 107. Mockup menú principal Blitz Party .....	111
Figura 108. Pantallas de juego .....	112

Figura 109. Herramientas de los minijuegos .....	112
Figura 110. Escenario de Binky Pursuit .....	114
Figura 111. Interfaz de Binky Pursuit. ....	114
Figura 112. Zorros de Binky Pursuit .....	115
Figura 113. Binkies de Binky Pursuit.....	115
Figura 114. Escenario de Whack-A-Mole .....	116
Figura 115. Interfaz de Whack-A-Mole .....	117
Figura 116. Martillo de Whack-A-Mole.....	117
Figura 117. Topo normal de Whack-A-Mole .....	117
Figura 118. Topo dorado de Whack-A-Mole .....	117
Figura 119. Zoomy de Whack-a-Mole .....	118
Figura 120. Escenario de Cowboy Duel .....	119
Figura 121. Interfaz de Cowboy Duel .....	119
Figura 122. Personajes de Cowboy Duel.....	120
Figura 123. Calendario de trabajo .....	124





# 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 videojuego 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, descrito en esta memoria.
- Jose Francisco Gómez Alemany: desarrollo de las mecánicas del videojuego.
- Marc Garcia Ferrer: desarrollo del apartado *back-end* del videojuego.
- 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.
- Diseñar y desarrollar la parte *front-end* del videojuego, creando una interfaz que, sin ser visualmente compleja, contenga una funcionalidad completa.
- Seguir un procedimiento organizado y estandarizado para el desarrollo del *front-end*, con fases dedicadas al análisis de requisitos, diseño, implementación y pruebas.
- Aplicar los principios de diseño de interfaces gráficas de usuario en la implementación del *front-end*.



## 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 la sección siguiente se explica en detalle cómo se ha implementado la parte *front-end* del videojuego; es decir, la interfaz de la aplicación y todas sus funcionalidades. Para ello, se procede a separar la explicación en distintos bloques: análisis, diseño, implementación y resultados del producto.

Por último, pero no menos importante, en las secciones finales 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.

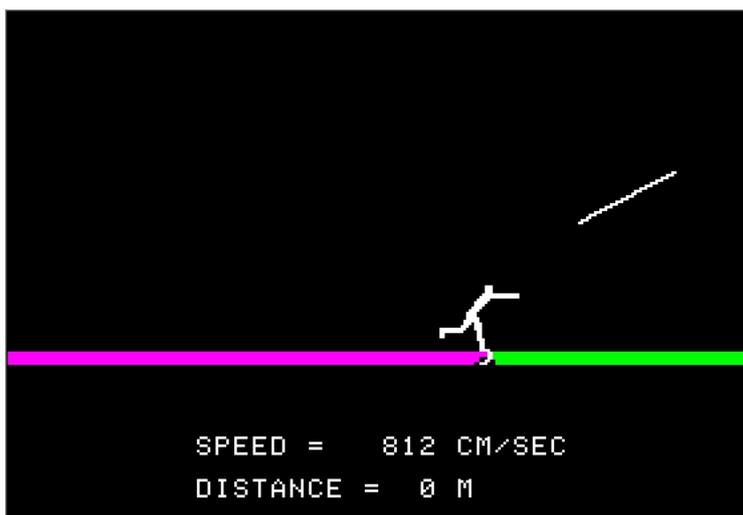


Figura 3. Lanzamiento de jabalina en *Olympic Decathlon*

El género no recibió entregas destacables hasta finales de los años 90, momento en el que salieron al mercado 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 estando juntos en el mismo sitio. Por ejemplo, poniendo bombas a otros jugando una partida 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.

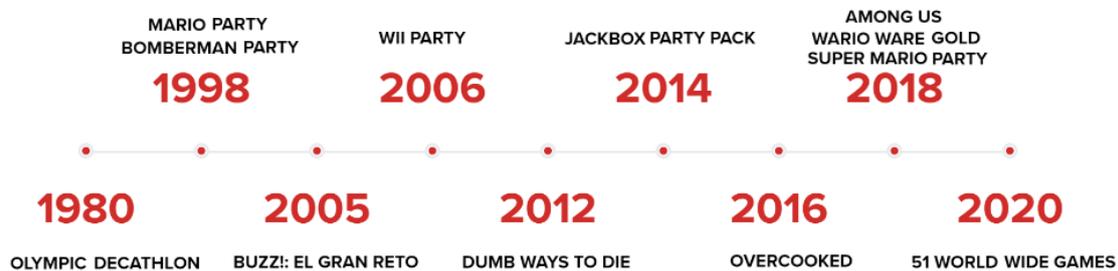


Figura 7. Cronología juegos *party* o de minijuegos

## 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), Gratuito (IOS y Android), 4,29€ (Switch)	Gratuito
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*





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 los 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.

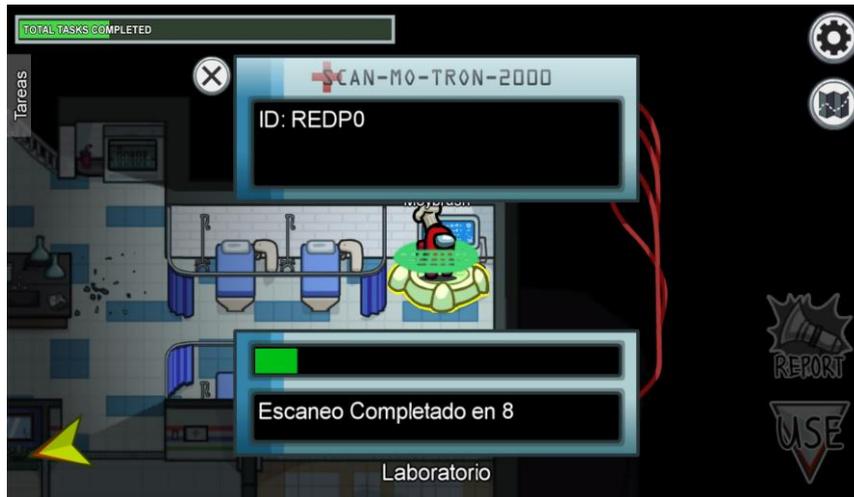


Figura 17. Jugador realizando una tarea en Among Us

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].

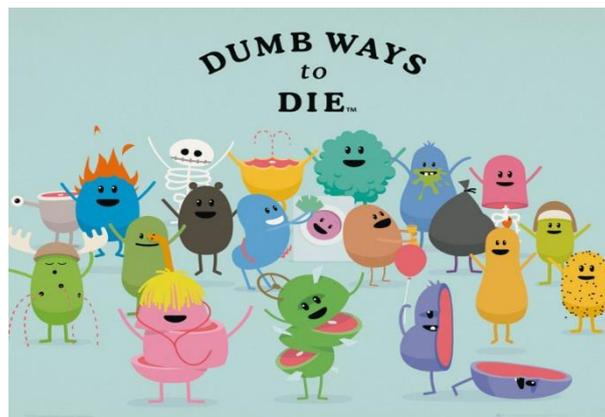


Figura 18. Portada de *Dumb Ways to Die*

*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*

Asimismo, 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.

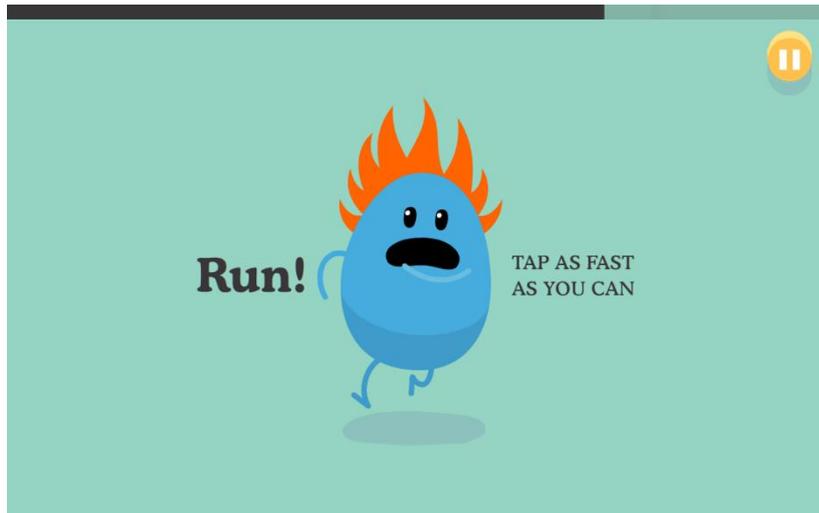


Figura 20. Microjuego de *Dumb Ways to Die*

### 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
Datos sobre 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>	<b>20000</b>	<b>19000</b>	<b>17000</b>
<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>	<b>2,500 €</b>	<b>2,375 €</b>	<b>2,125 €</b>
<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>	<b>40,660 €</b>	<b>35,660 €</b>	<b>35,660 €</b>
<b>Resultado Trimestral</b>	<b>-38,160 €</b>	<b>-33,285 €</b>	<b>-33,535 €</b>
<b>Resultado Trimestral Acumulado</b>	<b>-38,160 €</b>	<b>-71,445 €</b>	<b>-104,980 €</b>

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 el juego 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).



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

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	500000	400000	320000	250000	200000
12000	10000	20000	450000	700000	900000	1000000
16000	25000	520000	850000	1020000	1150000	1200000
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) relacionado 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.).

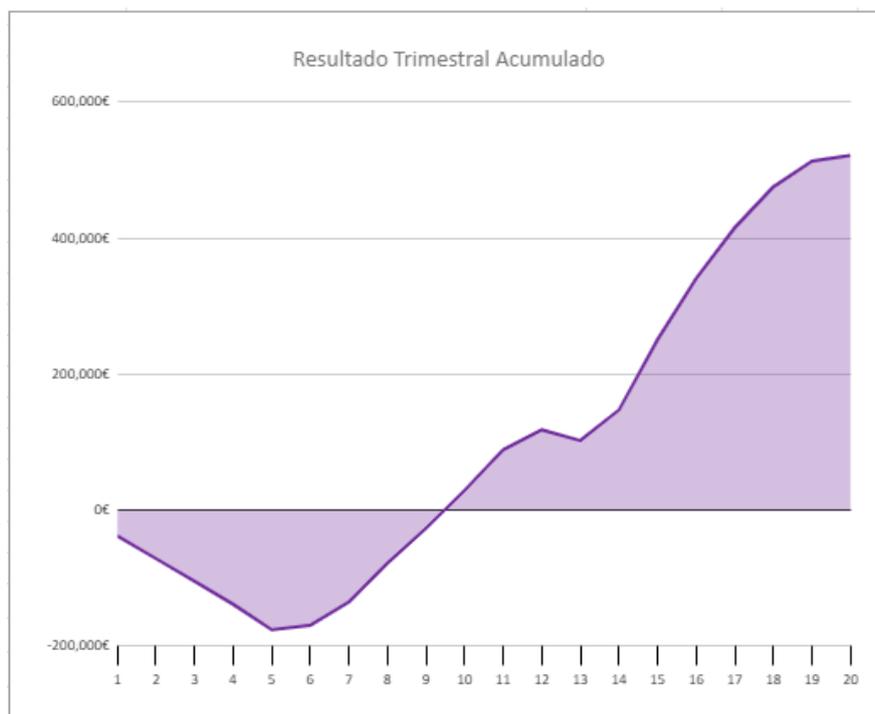


Figura 27. Resultado acumulado de ingresos

## 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>Problema</b></p> <p>En el mercado de juegos móviles, tanto los juegos estilo party como los juegos estilo minijuegos son un terreno bastante inexplorado.</p> <p>Algunos posibles competidores son Among Us, Wario Ware, Mario Party, Fall Guys o Clash Royale. Entre estos juegos, algunos son casuales, algunos son multijugador y algunos son estilo party. Pero ninguno de ellos cumple todas estas características.</p>	<p><b>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>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>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>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 smartphones o tablets.</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> <li>-Early adopters: amigos y familiares de los desarrolladores.</li> <li>-El modelo de negocio es Free to Play es decir el juego es gratuito pero contiene elementos cosméticos de pago que los jugadores podrán adquirir si lo desean.</li> </ul>
	<p><b>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>Canales</b></p> <ul style="list-style-type: none"> <li>- Medio de venta a través de Google Play Store y App Store.</li> <li>- Publicidad mediante redes sociales como Twitter o Instagram, Youtube y canales de comunicación tales como servidores de Discord.</li> </ul>	

## 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€

## 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 pueden 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.

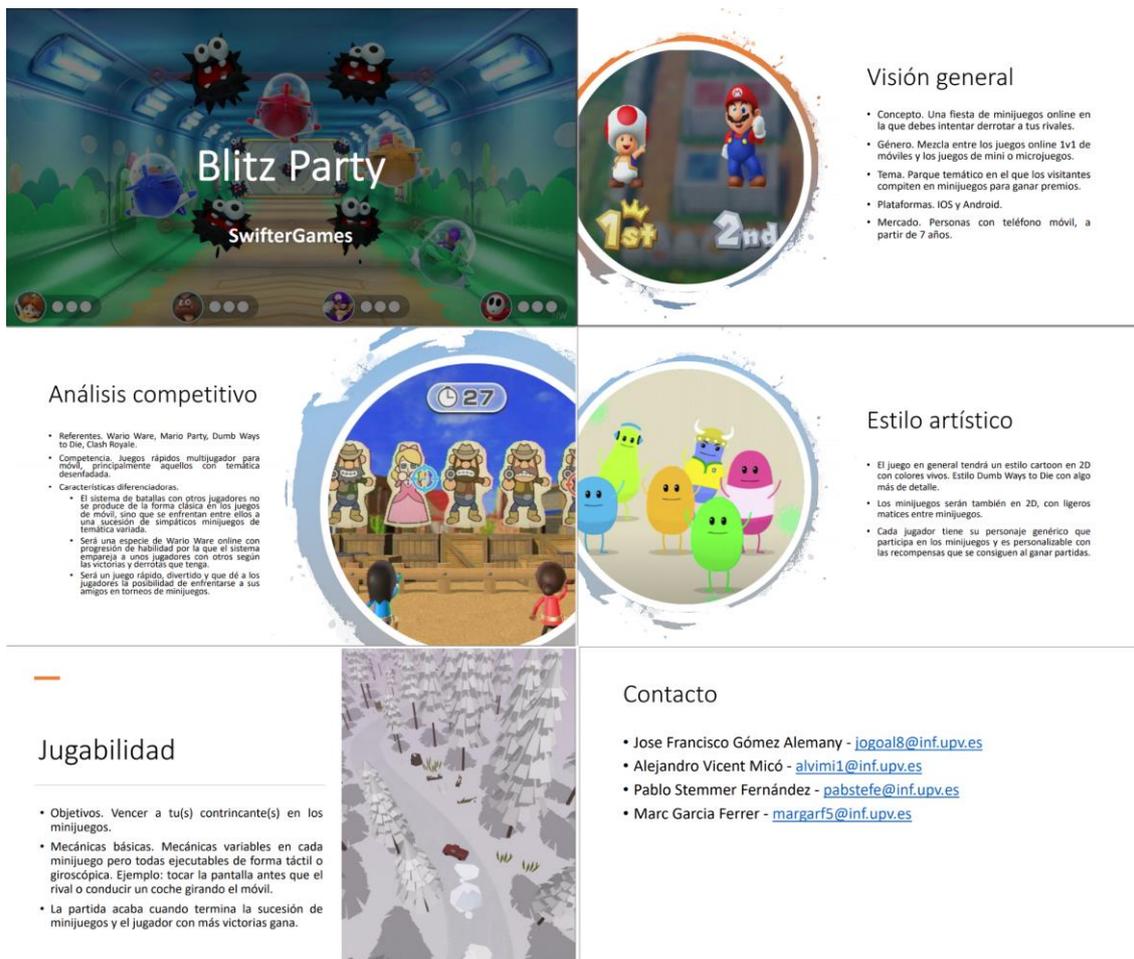


Figura 29. Pitch Doc

### 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 las respuestas cerradas, algunas en una escala del 1 al 10 y otras de respuesta abierta 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 más 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.

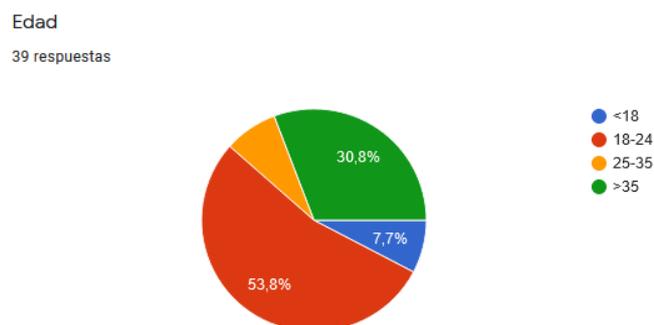


Figura 37. Edad de los participantes de la encuesta

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

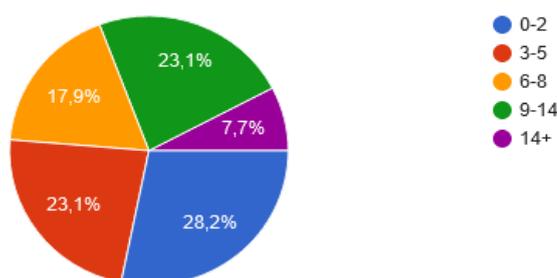


Figura 38. Tiempo que dedican a videojuegos los participantes de la encuesta

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.

¿Sueles jugar a videojuegos en dispositivos móviles?

39 respuestas

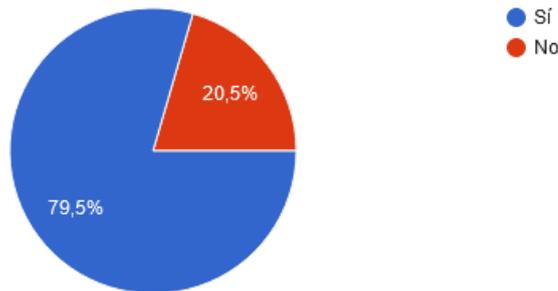


Figura 39. Porcentaje de participantes que juegan en dispositivos móviles

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

33 respuestas

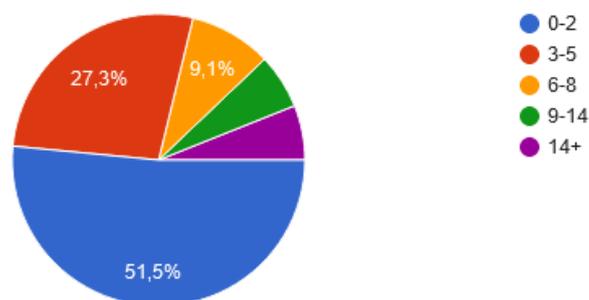


Figura 40. Horas que juegan los participantes de la encuesta en dispositivos móviles

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

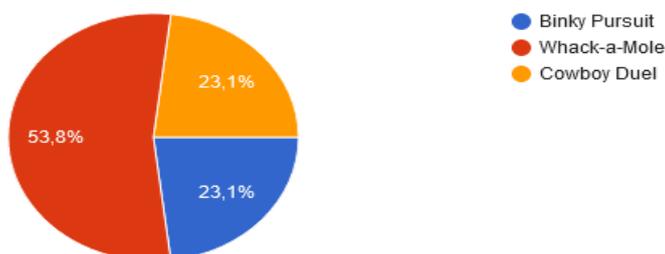


Figura 41. Minijuego favorito de los encuestados

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

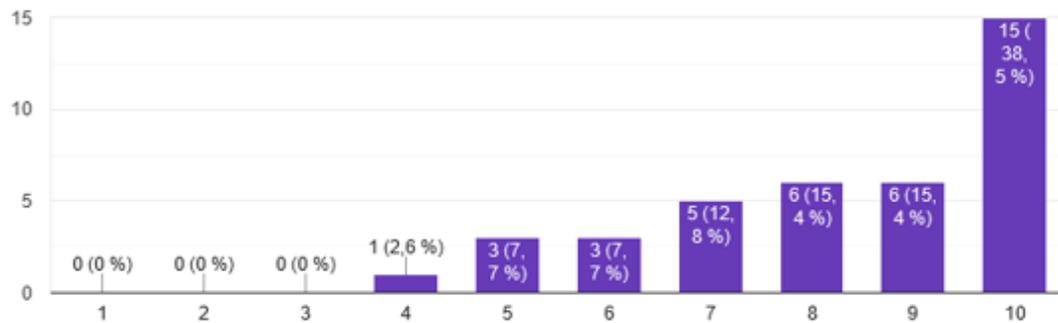


Figura 42. Puntuaciones de la curva de aprendizaje

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

39 respuestas

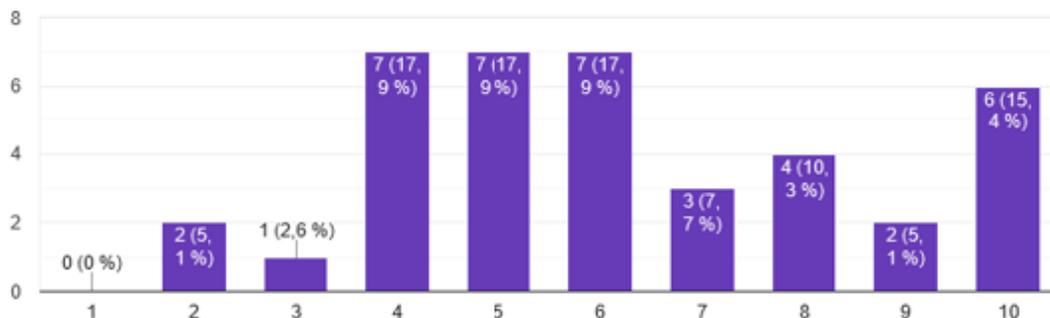


Figura 43. Puntuaciones sobre la dificultad del juego

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

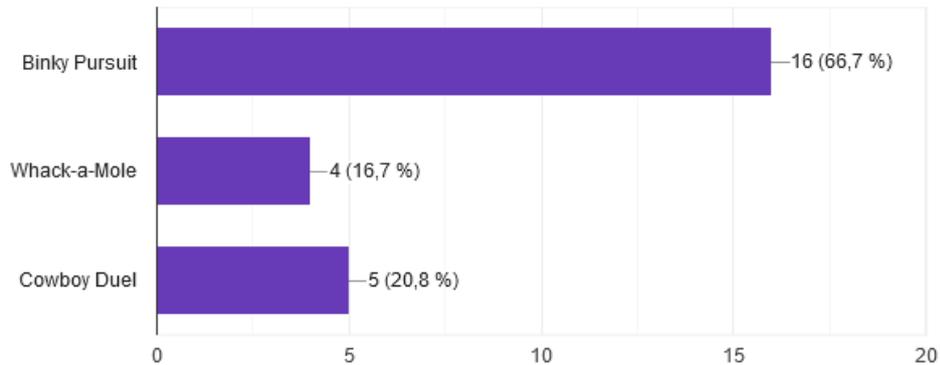


Figura 44. Opinión sobre dificultad excesiva de algún minijuego

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

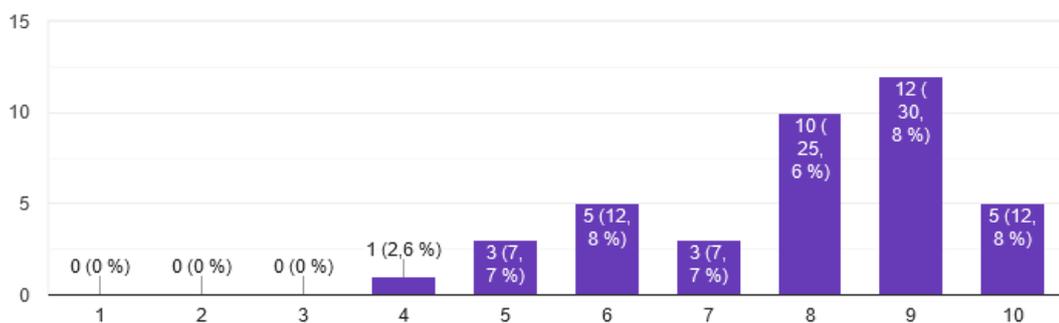


Figura 45. Opiniones sobre la claridad de la interfaz gráfica

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

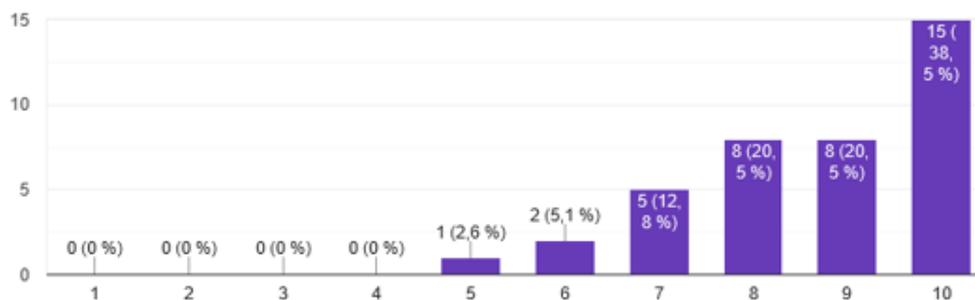


Figura 46. Opiniones sobre diseño y ambientación

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

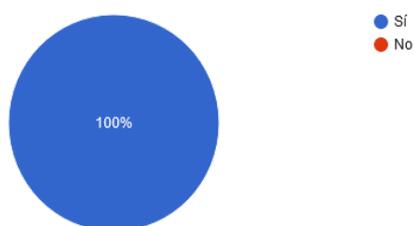


Figura 47. Opinión sobre la música de los minijuegos

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.

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

39 respuestas

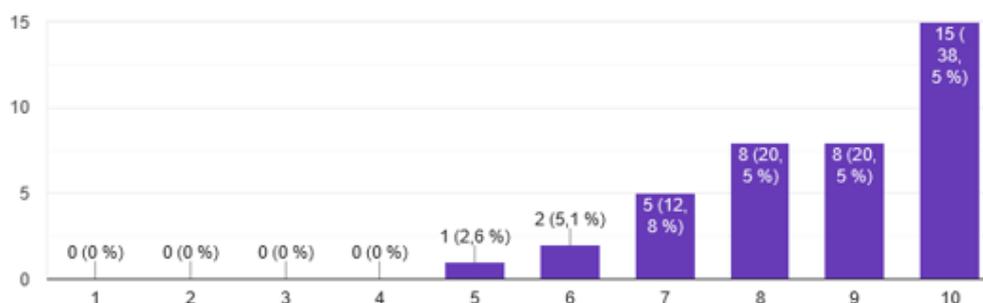


Figura 48. Puntuaciones generales de *Blitz Party*

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.

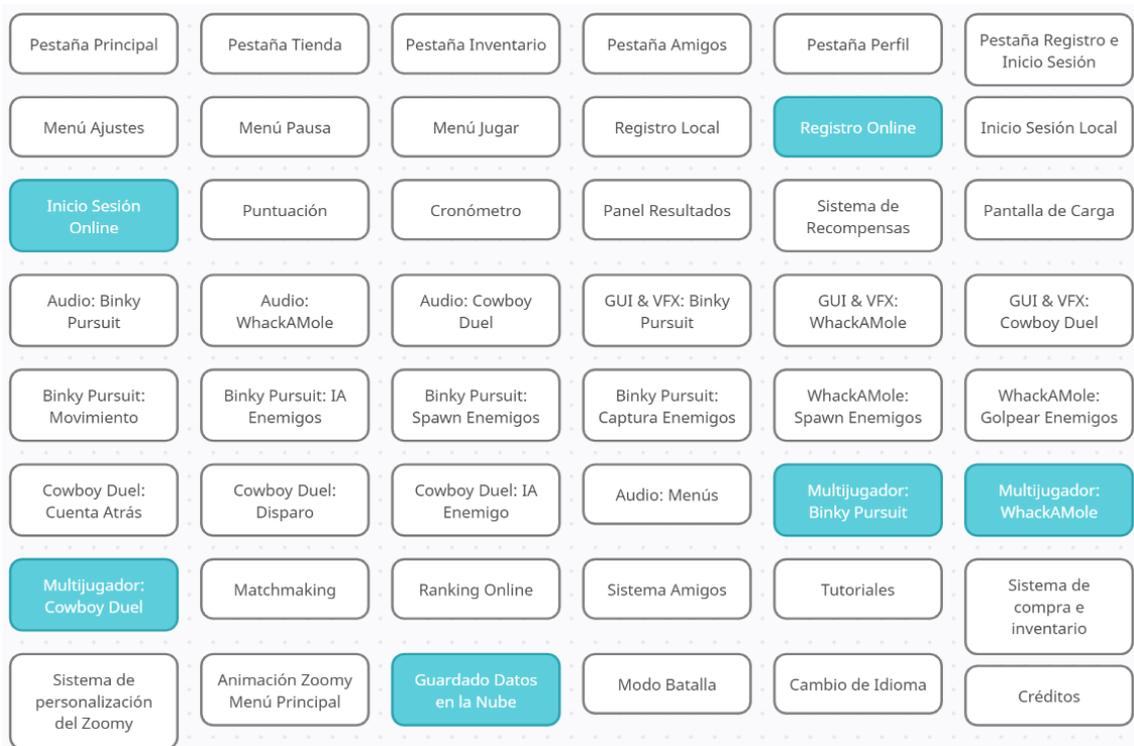


Figura 49. Mapa de características del segundo MVP

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.

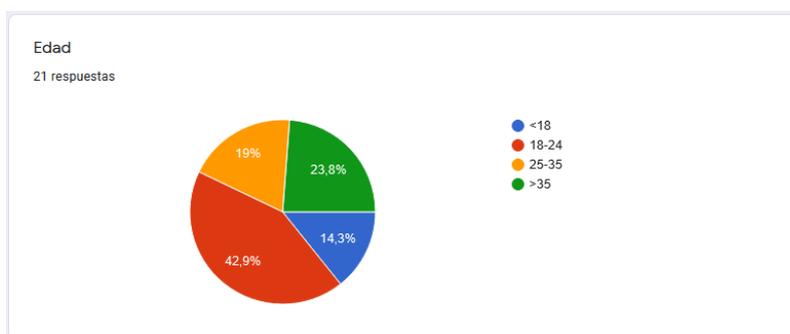


Figura 52. Pregunta rango de edad

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.

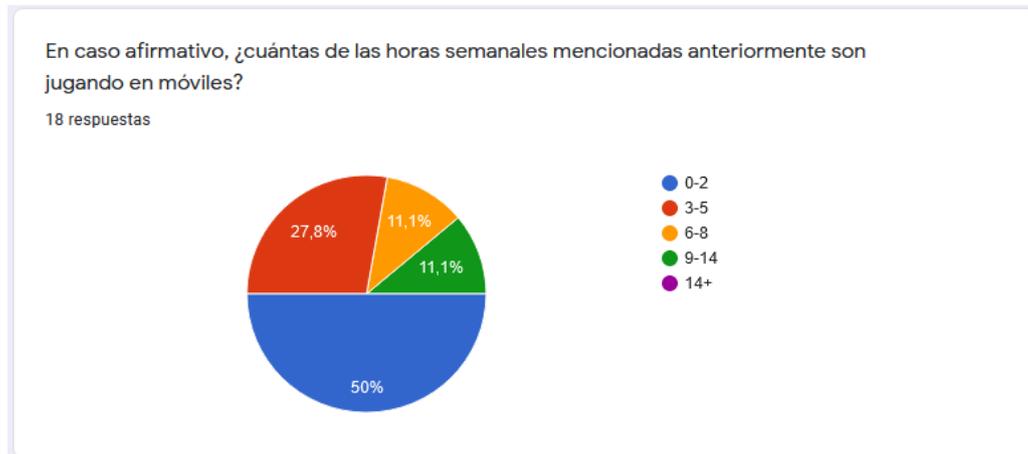


Figura 54. Pregunta horas jugadas en dispositivos móviles

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.

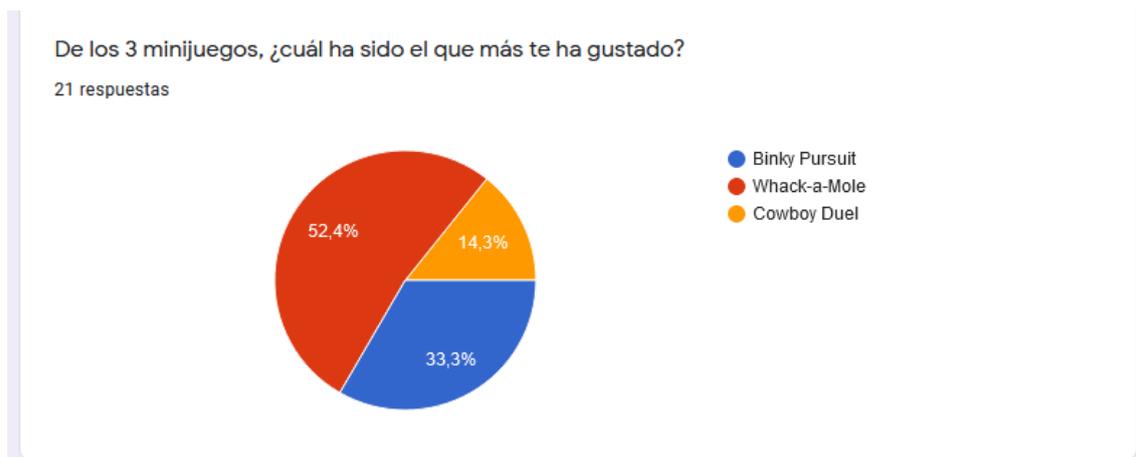


Figura 55. Pregunta minijuego preferido

Si comparamos los datos de la Figura 41 con los de la Figura 55, observamos que *Whack-A-Mole* 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.

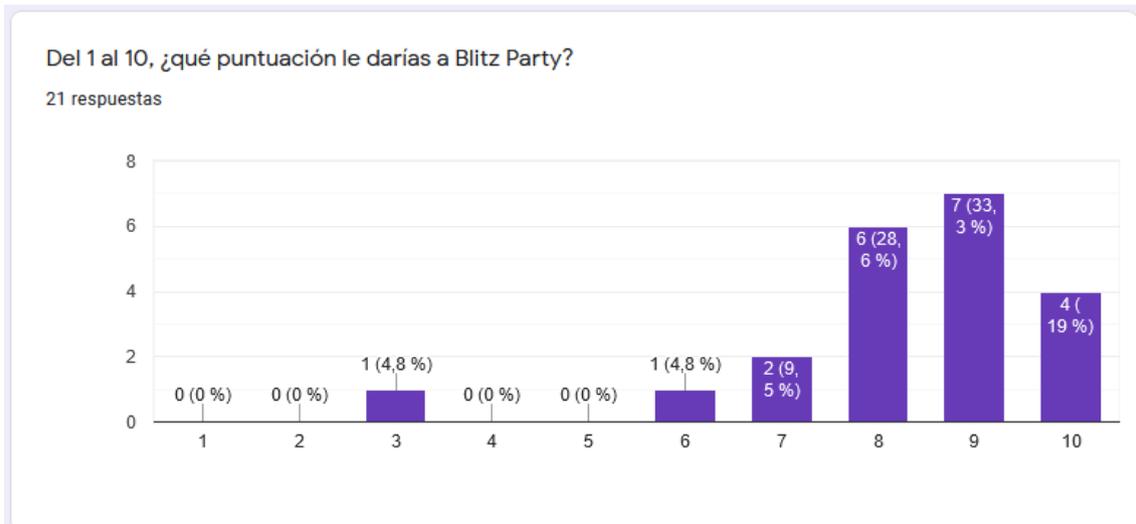


Figura 56. Pregunta puntuación general *Blitz Party*

En las puntuaciones generales del videojuego (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 Despliegue del producto y *marketing*

El producto se ha desplegado 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 aprecia en la Figura 58.

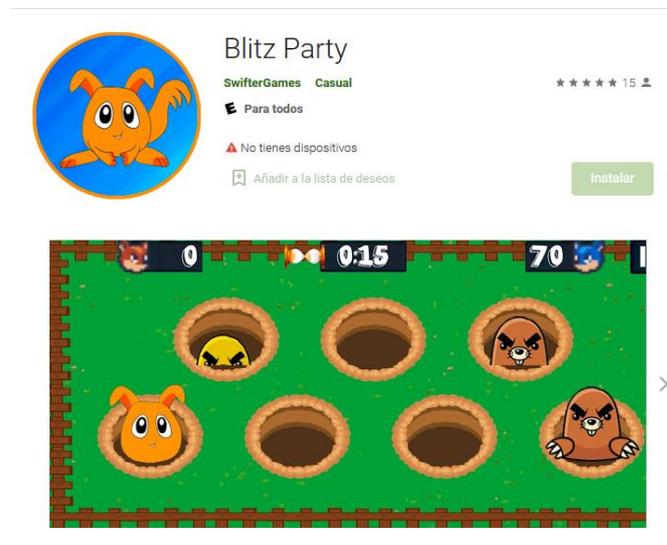


Figura 57. *Blitz Party* en la *Play Store*

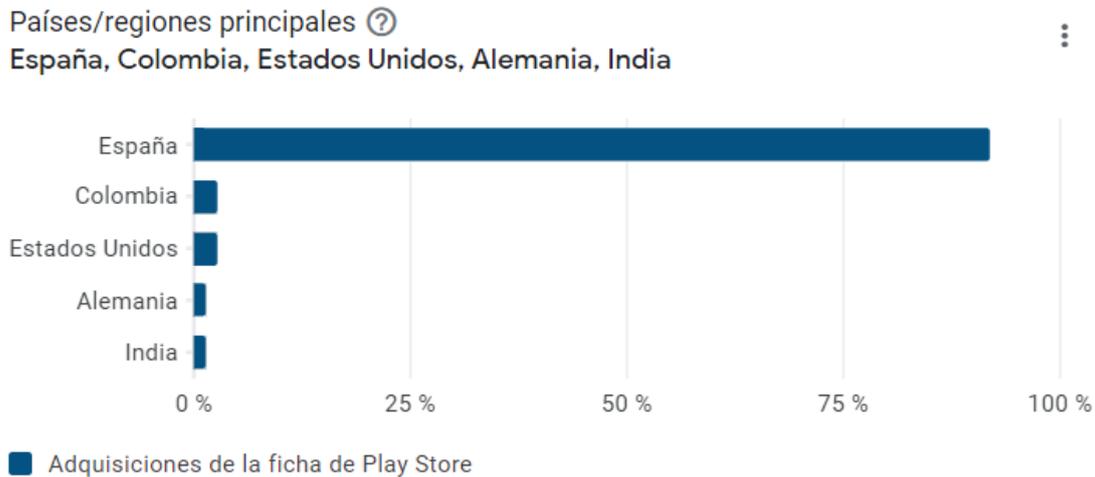


Figura 58. Países desde los que se ha descargado *Blitz Party*

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 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 622), 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 *front-end*

---

### 5.1 Introducción

En los próximos capítulos se va a explicar el trabajo realizado para la implementación del *front-end* del videojuego. El *front-end* es todo el apartado visual de una aplicación, que se centra en captar las interacciones del usuario, ya sea mediante teclado y ratón, mando o pantalla táctil, e interpretarlas para que el programa realice las acciones deseadas. El desarrollo se ha centrado en los siguientes puntos:

- Diseñar y programar las distintas pantallas del videojuego, desde la ventana de registro e inicio de sesión hasta las pestañas que forman parte del menú principal, como la tienda o el inventario. Estas pestañas son las que nos permiten navegar por la aplicación y acceder a todas las opciones de que dispone.
- Crear la interfaz gráfica de los minijuegos que conforman la aplicación; es decir, todos los elementos que vemos en la pantalla cuando estamos dentro de un minijuego: las puntuaciones, el cronómetro, la ventana de ajustes y la de resultados, etc.
- Llevar a cabo los dos puntos que se acaban de mencionar siguiendo una estética *cartoon*, amigable y fácil de entender, que esté en consonancia con el resto de la aplicación. Se ha prestado especial atención a que la interfaz de juego sea, no solo atractiva visualmente para los jugadores, sino también intuitiva y sencilla, con una curva de aprendizaje suave que permita a usuarios de todas las edades disfrutar de la experiencia de juego tan pronto como sea posible. La importancia de esta tarea radica en el hecho de que todos estos objetivos se han de realizar sin reducir las opciones ofrecidas, teniendo una funcionalidad completa al final del proceso de desarrollo.

Se ha decidido plantear el desarrollo *del front-end* en los siguientes apartados, característicos de cualquier proyecto informático:

- Análisis de requisitos y diseño de la solución.
- Implementación del producto.
- Pruebas y resultados obtenidos.

## 5.2 Metodología

Como se ha mencionado en el apartado 1.3, la implementación del videojuego se ha llevado a cabo siguiendo metodologías ágiles. El proceso de desarrollo está delimitado por dos MVP y dividido en cuatro *sprints*, unos periodos de tiempo dedicados al desarrollo del producto, con una duración de entre una y cuatro semanas. La estructura que se ha seguido se esquematiza en la Figura 67:



Figura 67. Línea de tiempo proceso de desarrollo

Los *sprints* uno, dos y tres se realizaron durante la asignatura de Entorno de Desarrollo de Videojuegos, presentando como trabajo final la versión del juego que se utilizó para el primer MVP. Después, se inició el *sprint* 4, en el que se desarrolló la versión final de la aplicación para el segundo MVP.

El procedimiento ágil de desarrollo consta de una serie de pasos que se repiten para cada uno de estos *sprints*. A continuación, se detallan los pasos seguidos como programador del *front-end* a lo largo del desarrollo.

Antes de empezar el proceso de desarrollo, se realiza una lista, llamada *backlog* (Figura 68), con todas las funcionalidades que se esperan implementar. Las funcionalidades, llamadas UT (unidades de trabajo), deben ser aspectos concretos del producto, de forma que se puedan realizar en un periodo de tiempo de entre cinco y 20 horas aproximadamente. Algunos ejemplos de UT podrían ser la programación del menú de ajustes o de la pantalla de carga. Si una UT es muy compleja y tiene una carga de trabajo de muchas horas, se le llama épica y se recomienda dividirla en UT más pequeñas para que el desarrollo sea más dinámico. Por ejemplo, el menú principal sería una UT épica que se podría dividir en tienda, inventario, ventana central, etc.

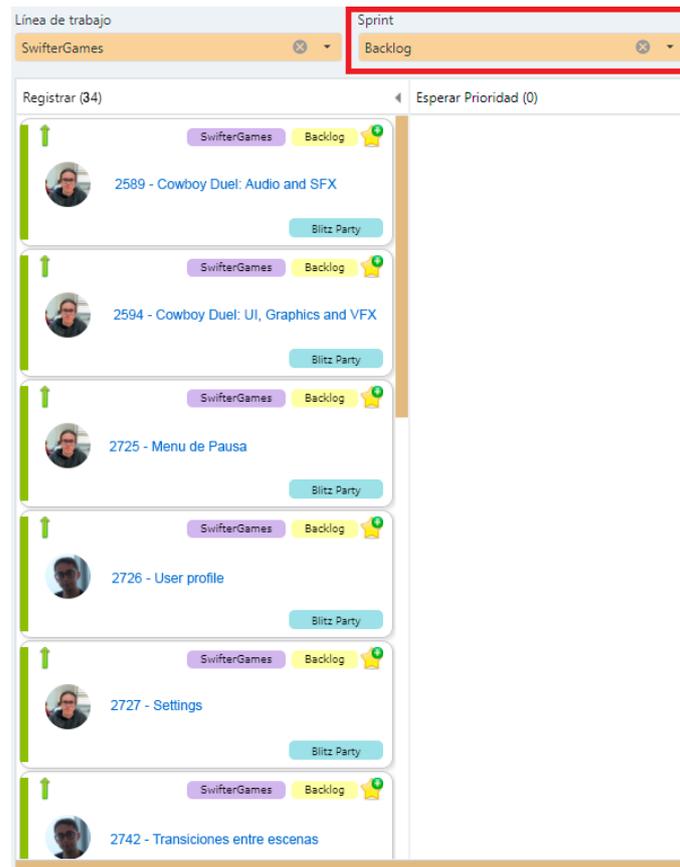


Figura 68. *Backlog*

Cuando el *backlog* ya está hecho, se decide qué UT se van a realizar en cada *sprint*. Se estima la cantidad de horas que el equipo de trabajo puede dedicar durante el *sprint*, y también el tiempo necesario para la implementación de las UT, a partir de los requisitos que se ha especificado para cada una de ellas. En función de estas variables, se escogen qué UT van a formar parte del *sprint* y se diseñan los prototipos de la interfaz de usuario.

Asimismo, y como se explicará posteriormente en el apartado 6.4, se establecen una serie de pruebas de aceptación (PA) para cada UT, que controlan un aspecto concreto de la funcionalidad que se va a implementar. Seguidamente, las UT se mueven a la columna de esperar *sprint*, y el día de inicio del *sprint*, pasarán a la columna programar (Figura 69), desde donde se irán seleccionando poco a poco y por orden de prioridad cuando un miembro del equipo esté libre. Esta persona se hará cargo de la UT y empezará su implementación. Cuando se haya terminado de implementar cada UT, se situará en la última columna, realizar pruebas, desde la que se deberá comprobar mediante las PA que el funcionamiento del código desarrollado es el esperado.

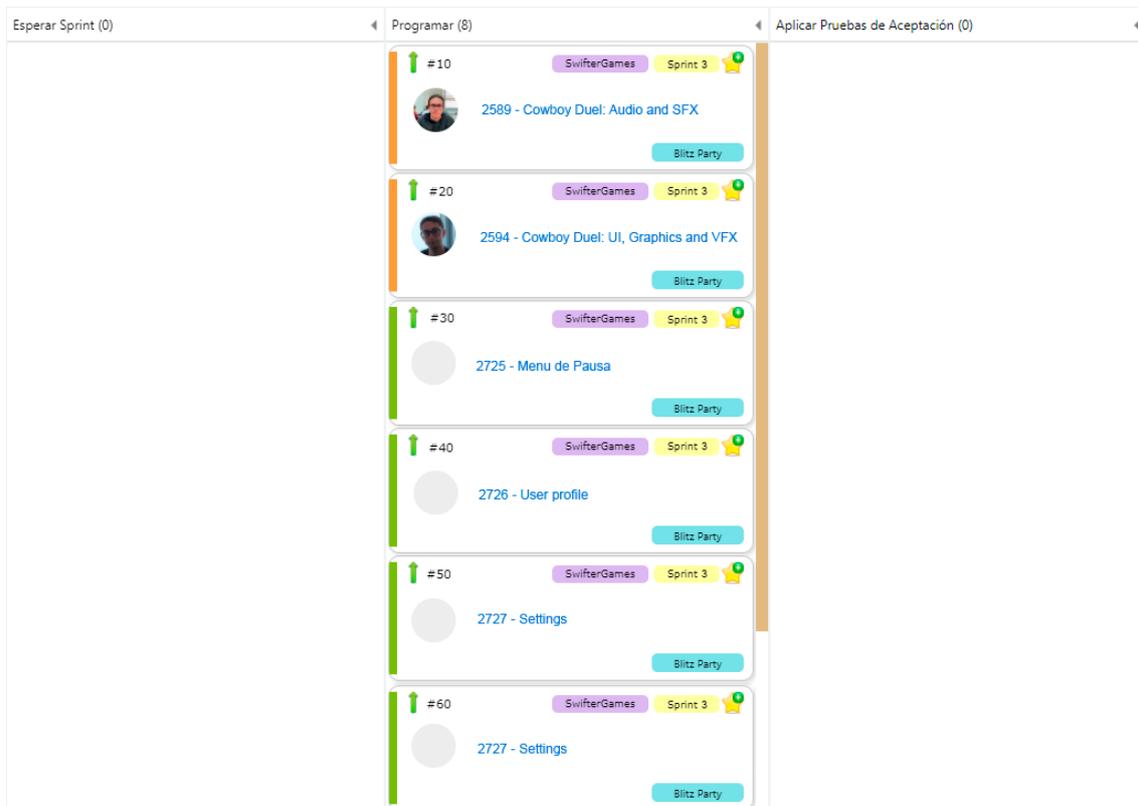


Figura 69. UT en programación

Además, el equipo de este proyecto ha realizado reuniones diarias de unos 10 minutos en las que se ha expuesto el trabajo realizado por cada miembro y cuáles iban a ser los siguientes pasos. Estas reuniones son de gran utilidad para todos los desarrolladores, ya que sirven para organizarse, saber cómo avanza el proyecto y ayudarse mutuamente en caso de que se haya encontrado algún impedimento.

Posteriormente, cuando acaba cada *sprint*, se realiza la *Sprint Review*, una reunión más extensa en la que se evalúan cuestiones como si la carga de trabajo para el *sprint* era demasiado alta o baja, si ha habido problemas de comunicación en el equipo o si el producto ha llegado al estado que se esperaba antes de empezar, entre otras. Para ello se analizan los datos disponibles en la plataforma *Worki*, que ayudan a entender cómo ha ido el *sprint* (Figura 70). Finalmente, se establece la estrategia que se va a seguir para el próximo *sprint*, teniendo en cuenta la información extraída de la *Sprint Review*.



Figura 70. Estadísticas del *sprint* en *Worki*

Cabe mencionar que los *sprints* cuya fecha de finalización coincide con uno de los MVP son todavía más importantes a la hora de fijar una estrategia y organizar al equipo de desarrollo para el siguiente *sprint*. Esto se debe a que el producto se publica en la *Google Play Store*, se realiza un experimento con los usuarios y se obtiene *feedback* gracias a los comentarios y las respuestas de los jugadores a una encuesta.

La opinión de los usuarios es igual o más importante que la del equipo de este trabajo ya que, además de ser el público al que va dirigida la aplicación, su visión del juego es externa al desarrollo y por tanto su punto de vista es objetivo y diferente. Este *feedback* recibido debe ser tenido en cuenta junto con las conclusiones extraídas de la *Sprint Review* para marcar los objetivos a seguir, introducir cambios y mejoras según las opiniones de la comunidad y corregir posibles errores encontrados gracias a las pruebas realizadas por todas las personas que han descargado el juego.

# 6. Análisis de requisitos

---

## 6.1 Ingeniería de requisitos

Para la construcción de una aplicación que cumpla con las expectativas del cliente se deben entender cuáles son sus necesidades, de forma que se puedan expresar formalmente para que los desarrolladores las implementen. Este proceso es conocido como ingeniería de requisitos y engloba el conjunto de actividades destinadas a definir las características de un sistema software que satisfaga las necesidades de clientes y usuarios.

Aunque esta tarea pueda parecer sencilla, en realidad es todo lo contrario. Los clientes no conocen los lenguajes de programación que se van a utilizar para desarrollar la aplicación y por tanto especifican sus requisitos utilizando lenguaje natural. Esta especificación suele ser ambigua y los clientes se desentenden del equipo de desarrollo una vez superado este punto. Por esta razón, es muy alta la cantidad de aplicaciones que se desestiman unos meses después de iniciar su desarrollo debido a que no cumplen con las expectativas de los clientes.

Aunque abandonar el proyecto es un caso extremo, también existe la posibilidad de que, después de reunirse con los *stakeholders*, se tenga que cambiar el código para que se adapte mejor a sus necesidades. Esto supone una serie de costes asociados a la reparación de estos errores y la implementación de cambios, por lo que al equipo de desarrollo le interesa realizar una buena ingeniería de requisitos y estar en contacto constantemente con los clientes mediante reuniones periódicas.

La ingeniería de requisitos se divide en elicitación, especificación, validación y gestión de requisitos, y consta de diversas técnicas que ayudan al programador a entender las funcionalidades que el cliente quiere introducir en su producto. El equipo de este TFG no cuenta con clientes que les planteen sus requisitos, así que ha entrevistado a familiares y amigos, que han desempeñado el papel de *stakeholders*. Asimismo, los miembros del equipo han hecho a la vez de desarrolladores y de clientes.

## 6.2 Elicitación de requisitos

La elicitación de requisitos es la primera tarea que se ha de desempeñar a la hora de realizar la ingeniería de requisitos. Consiste en determinar de dónde vienen los requisitos y cómo obtenerlos. Para ello, algunas de las técnicas más famosas utilizadas son las entrevistas, el *brainstorming*, los casos de uso o las historias de usuario. Al ser un videojuego y no tener clientes que satisfacer, el equipo de desarrollo decidió utilizar el *brainstorming* como técnica de elicitación de requisitos.



Además, para conocer la opinión de los usuarios sobre aspectos de la interfaz gráfica, se realizaron entrevistas con familiares y amigos.

### 6.2.1 *Brainstorming*

La sesión de *brainstorming* se dividió en tres fases. Primeramente, se determinaron los objetivos que se esperaba conseguir al final del proceso de desarrollo: un menú funcional para dispositivos móviles con diversas características y una serie de minijuegos variados, aptos para todas las edades. A continuación, se procedió a generar tantas ideas como fuese posible. En esta fase, la cantidad es más importante que la calidad, pues es de especial interés que todos los participantes expongan las funcionalidades que tienen en mente, sin importar si son difíciles de realizar o poco interesantes.

Esta forma de trabajar facilita la obtención de funcionalidades innovadoras y originales. Se deben apoyar todas las ideas expuestas y evitar la crítica., ya que estas se pueden descartar más adelante si no se consideran necesarias. Esta fase es de especial interés para la implementación del *front-end*, pues la mayoría de las características de los menús y de la interfaz de los minijuegos se originan aquí (Figura 71, ideas en gris).



Figura 71. Ideas *brainstorming*

En la última fase, se procede a evaluar todas las ideas expuestas durante la sesión, organizándolas en apartados para mayor claridad. Algunas de estas ideas pueden reformularse si, después de debatirlo, la forma de entender dicha idea ha cambiado. Además, otras ideas pueden combinarse si por sí solas no son lo suficientemente importantes (en el caso de la pantalla de carga, que

combina la barra de carga y la animación) o separarse en otras tareas más específicas (como los minijuegos, que se dividen en apartados). Finalmente, se priorizan las que se van a implementar y se descartan las que no. En el epígrafe 3.1 se muestra, mediante un mapa de características, cuál ha sido el resultado final, con todas las funcionalidades que se espera implementar.

### 6.2.2 Entrevistas

Como método auxiliar al *brainstorming* realizado por el equipo de desarrollo y para conocer la opinión de personas externas a este, se llevaron a cabo entrevistas con familiares y amigos de distintas edades y gustos. Al principio de las entrevistas se explicó cuál iba a ser el producto a implementar: un videojuego estilo *party* para dispositivos móviles, con minijuegos y la posibilidad de jugar contra otras personas. Después de cerciorarse de que el entrevistado había entendido el concepto inicial, se procedió a realizar diversas preguntas sobre temas tanto generales como más específicos.

En lo que respecta al apartado *front-end* (menús del juego, interfaz gráfica, etc.), destacan las siguientes preguntas. Primero, se preguntó la opinión de los usuarios sobre el personaje del juego. Se enseñó un diseño inicial y se obtuvo *feedback* acerca de qué partes cambiar. Algunas de las respuestas (“Se parece demasiado a *Kirby*” o “Le haría las orejas más grandes”) ayudaron a diseñar el personaje que se utiliza en la versión actual del videojuego (Figura 72).

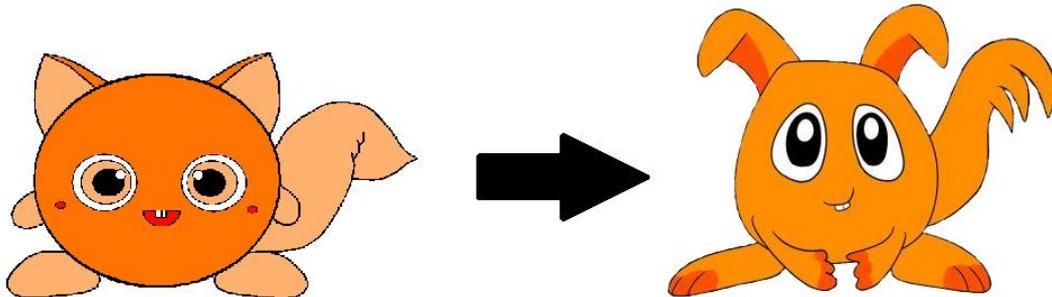


Figura 72. Evolución del personaje

Por otra parte, las demás preguntas de la entrevista trataban temas relacionados con la interfaz gráfica y los menús del juego. Se preguntó a los entrevistados qué funcionalidades les gustaría que formasen parte de la aplicación y las respuestas fueron variadas. Algunas de estas respuestas son las siguientes:

- “Una tienda de objetos estaría bien.”
- “Los ajustes típicos de cambiar el volumen.”
- “Que se pueda escoger contra quien juegas”
- “Poder cuidar de mi personaje como en el *Pou*.”
- “Poder agregar amigos y hablar con ellos.”
- “Cambiar el color del personaje.”

- “Que sea fácil de jugar”
- “Unirse a un clan y competir entre clanes.”
- “Sistema clasificatorio con ranking”
- “Que se pueda jugar contra la máquina, que soy muy malo.”

Algunas de estas respuestas ofrecen ideas sobre ventanas que se pueden diseñar, como por ejemplo una ventana de tienda u otra de personalización de personaje, para equipar los objetos de la tienda o cambiar el color del personaje. Otras señalan características importantes para los usuarios, como que la dificultad del juego no sea elevada o que se pueda interactuar con nuestro personaje. Todas estas características se tuvieron en cuenta más tarde a la hora de diseñar los *mock-ups* del menú principal.

### 6.3 Especificación de requisitos

Para que el equipo de desarrollo pueda implementar satisfactoriamente las funcionalidades seleccionadas en la fase de elicitación, se deben especificar. Mediante la especificación de requisitos se pretende estructurar coherentemente las tareas a realizar usando un lenguaje claro y sin ambigüedades.

Para ello se ha utilizado la herramienta *Worki Process* que, como se ha explicado en el epígrafe 5.2, es clave a la hora de organizar el trabajo y seguir una metodología ágil. En el tablero *Kanban* de *Worki* hay dos columnas importantes situadas entre la elicitación de requisitos y su implementación: especificar requisitos y diseñar y estimar (Figura 73).

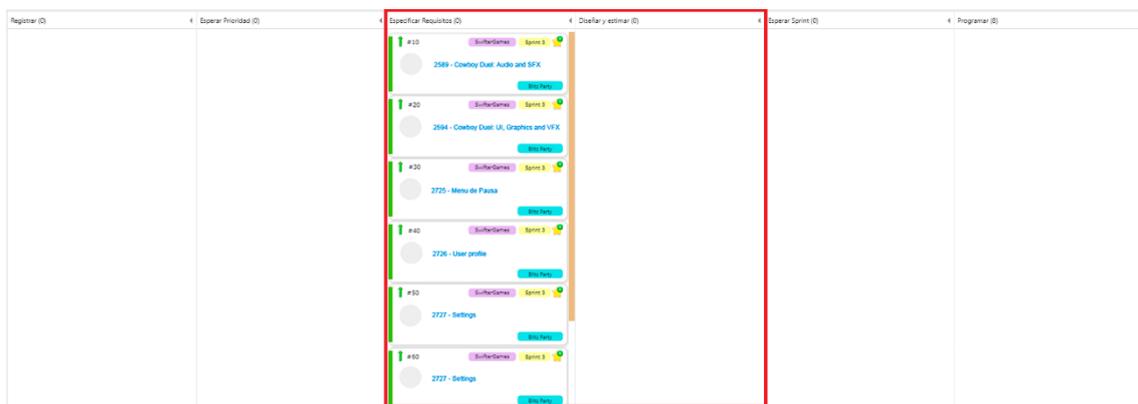


Figura 73. Columnas de especificar requisitos y diseñar

En la columna de especificar requisitos se explica qué es lo que cada funcionalidad o UT debería realizar y cómo. Para ello, se escribe una descripción en la que se detallan las características de la UT y se preparan las pruebas de aceptación pertinentes. Asimismo, en la siguiente columna se respalda la especificación de requisitos mediante *mock-ups*, que ayudan al desarrollador a entender cómo se tiene que diseñar e implementar la interfaz.

Los requisitos correspondientes al *front-end* que se han especificado en cada *sprint* son los siguientes:

- Sprint 1:
  - Interfaz del minijuego *Binky Pursuit*, con un cronómetro, las puntuaciones de cada jugador y un menú con el resultado final.
  - Pestaña central del menú principal, con los botones necesarios para desplazarse por todas las pestañas del menú principal.
  - Menú de selección de minijuegos.
- Sprint 2:
  - Interfaz del minijuego *Whack-A-Mole*, con un cronómetro, las puntuaciones de cada jugador y un menú con el resultado final.
- Sprint 3:
  - Menú de ajustes, con la posibilidad de modificar el volumen de la aplicación.
  - Menú de pausa, accesible desde los minijuegos, con la posibilidad de modificar el volumen dentro del minijuego y la opción de salir al menú principal.
  - Pestaña de la tienda, con la funcionalidad de comprar objetos si el jugador tiene suficientes bellotas.
  - Pestaña del inventario, desde la que se administrarán los accesorios comprados en la tienda.
  - Pestaña del perfil de usuario, con estadísticas sobre sus partidas.
  - Modo batalla, en el que se juega de forma seguida a tres minijuegos aleatorios.
  - Interfaz del minijuego *Cowboy Duel*, con una cuenta atrás, las puntuaciones de cada jugador y un menú con el resultado final.
  - Pantalla de carga, al principio de cada minijuego, con una animación de los zorros corriendo.
- Sprint 4:
  - Adaptación de las interfaces de usuario de los minijuegos al modo multijugador.
  - Ventana de búsqueda de jugadores, que se muestra al inicio de los minijuegos multijugador.



## 7. Diseño de la solución

### 7.1 Mock-ups

Los *mock-ups* son una maqueta de la interfaz gráfica de usuario de una aplicación, en una versión parcial y preliminar. Pueden diseñarse en papel, en una pizarra o incluso utilizando herramientas de software que facilitan su creación. Se dice que son parciales y preliminares porque diseñar un *mock-up* no implica que su versión final sea exactamente igual. Todo lo contrario, los *mock-ups* se conciben con el objetivo de mostrar a los clientes y al resto de desarrolladores, mediante la técnica del prototipado, cuál es la idea de diseño inicial para una interfaz.

Al contar con referencias gráficas y no solo con descripciones y texto, los usuarios son capaces de entender mejor cómo funcionará la aplicación, y de esta forma dar *feedback* a los desarrolladores sobre qué funcionalidades les gustaría introducir y cuáles van a apartarse por el momento. Así, la idea inicial sufrirá modificaciones y la interfaz final acabará siendo distinta a la primera versión enseñada.

Después de difundir el primer cuestionario sobre la interfaz gráfica y obtener *feedback* de todas las personas que lo contestaron, se procedió a realizar unos *mock-ups* iniciales que recogían las funcionalidades más demandadas por los usuarios, así como las más importantes según el equipo de desarrollo. Como se observa en la Figura 74, se decidió dividir el menú principal en 5 pestañas, navegables mediante un grupo de botones situados en la parte inferior de la pantalla: tienda, personalización de personaje, pestaña principal o de juego, amigos y perfil de usuario.

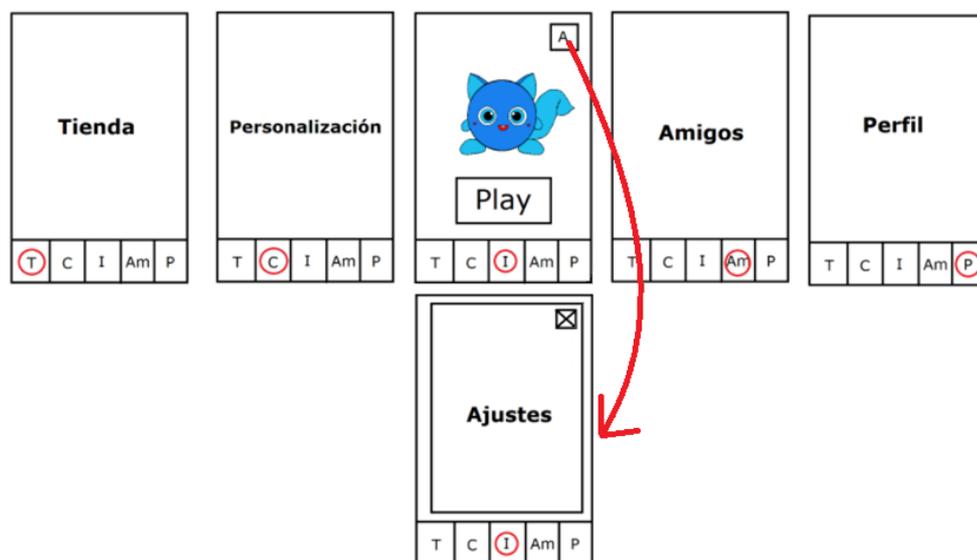


Figura 74. *Mock-up* menú principal

Algunas de las funcionalidades más pedidas por los entrevistados, como la posibilidad de interactuar con nuestro personaje (el cual aparece en la pestaña central) o la presencia de un menú de ajustes, se añaden al *mock-up* para tenerlas en cuenta para su futura implementación. Asimismo, se procede también a diseñar un *mock-up* para el menú de selección de partida, cuyo resultado se ve en la Figura 75.

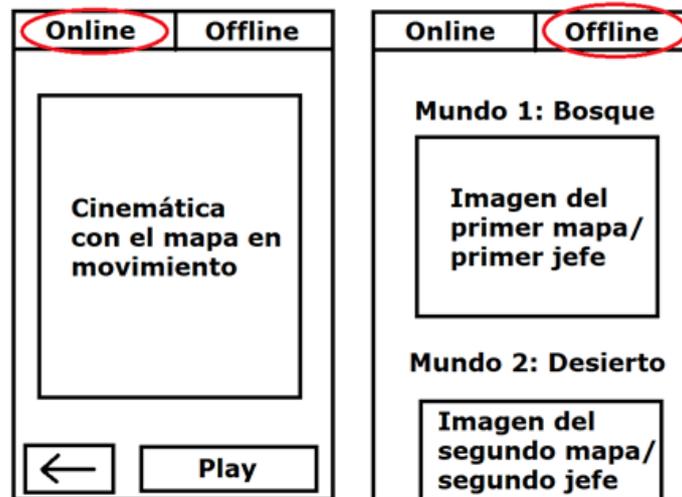


Figura 75. *Mock-up* selección de partida

Algunas ideas iniciales del equipo de desarrollo, concebidas durante la sesión de *brainstorming*, se introdujeron en el *mock-up* de selección de partida. Por una parte, los botones situados en la parte superior de la pantalla ofrecen la posibilidad de jugar tanto *online* contra amigos como *offline* contra una IA. Además, la creación de zonas temáticas, cada una con sus minijuegos y su jefe final, se tuvieron en cuenta en el diseño del menú *offline*.

Una vez diseñados estos *mock-ups*, se procede a realizar diversas entrevistas con los amigos y familiares que participaron en la entrevista inicial, con la finalidad de enseñarles los *mock-ups* y preguntar su opinión al respecto. Después de obtener su aprobación, los *mock-ups* se adjuntan a las unidades de trabajo de *Worki* que les corresponden (Figura 76), de forma que la persona encargada de implementar dicha UT sea capaz de entender qué se espera de ella al final de su desarrollo.

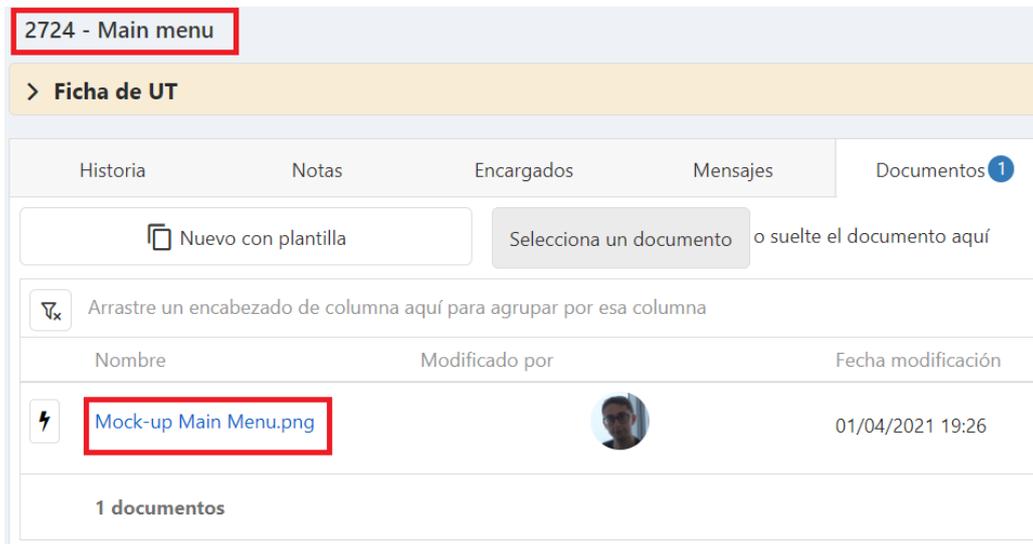


Figura 76. Worki: Mock-up menú principal

No obstante, y como se ha explicado en este epígrafe, el diseño mostrado en la Figura 75 no corresponde con la interfaz final del videojuego, pues el menú de selección de partida ha sido modificado, ya sea por falta de tiempo o porque se ha considerado adecuado realizar cambios. Estas modificaciones no son en ningún momento negativas ya que, al seguir una metodología ágil, el equipo de desarrollo es capaz de valorar al inicio de cada *sprint* las funcionalidades que se van a implementar y, en caso de no estar de acuerdo con el *mock-up* diseñado inicialmente, realizar los cambios pertinentes para que al final del *sprint* se obtenga un MVP con unas funcionalidades completas.

## 7.2 Principios del diseño de interfaces

Un diseño efectivo de las interfaces de usuario proporciona muchas ventajas y puede tener un gran impacto en la sociedad: los médicos pueden realizar diagnósticos más precisos, los niños pueden aprender más rápido, etc. Por otro lado, si la interfaz está mal diseñada y no es intuitiva, puede conllevar la aparición de errores o la frustración del usuario. Por esta razón, es de especial importancia prestar atención al diseño de la interfaz y de los menús de un videojuego, pues de ellos puede depender el éxito o el fracaso del producto final.

Una interfaz debería ser efectiva, eficiente y capaz de satisfacer a los usuarios. Efectiva en el sentido de que cumple con las funcionalidades para las que ha sido diseñada; es decir, los usuarios son capaces de utilizarla para jugar y navegar entre los distintos menús del videojuego. Eficiente porque el tiempo necesario para completar las tareas que se acaban de mencionar ha de ser reducido, así como el tiempo dedicado a aprender cómo funciona la aplicación. Y satisfactoria porque los controles y el diseño deben ser cómodos, bonitos y agradables para los usuarios, ya que de ello depende que sigan utilizando el producto.

Para diseñar *Blitzs Party*, se han seguido los cuatro principios psicológicos del diseño de interfaces:

- **Los usuarios ven lo que esperan ver:** se basa en mantener una interfaz consistente que ayude al usuario a orientarse. Para ello y como se puede apreciar en la Figura 77, se mantiene un mismo esquema cromático en toda la aplicación, en el que predominan los colores azul y amarillo, reservando el color blanco para los textos.

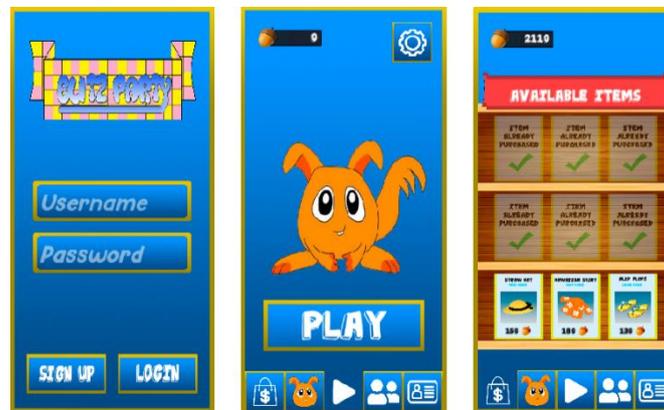


Figura 77. Consistencia en el esquema cromático

- **Los usuarios tienen problemas centrándose en más de una actividad a la vez:** los usuarios suelen dividir su atención en más de una actividad al mismo tiempo, por lo que la interfaz debería recordarles qué es lo que tienen que hacer a continuación. Por esta razón, deberían agruparse las cosas que van juntas y la información importante se tendría que colocar en una posición prominente. Siguiendo este principio, se han posicionado las alertas y menús emergentes en el centro de la pantalla (Figura 78), difuminando el fondo para que el jugador se centre solo en el menú que acaba de aparecer.

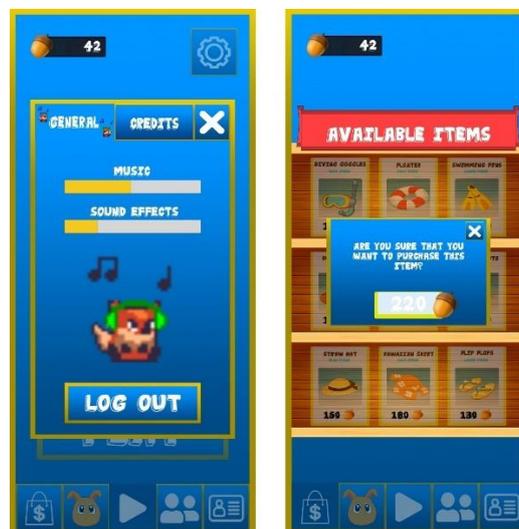


Figura 78. Menús emergentes

- **Es más fácil reconocer algo que recordarlo:** la interfaz gráfica no debe depender de la memoria del usuario, sino que ha de ser ella misma la que proporcione la información. Teniendo esto en cuenta, es un fallo de varias aplicaciones tener muchos botones cuyo único contenido es texto. En su lugar, una opción más acertada es que estos botones estén representados con un icono que ayude al usuario a entender cuál es su función con solo un vistazo. En *Blitz Party*, como se puede observar en la Figura 79, los botones siguen esta técnica y utilizan iconos orientativos cuyo significado es ampliamente conocido: una bolsa para la tienda, un carné para el perfil de usuario, un engranaje para el menú de ajustes, etc. Adicionalmente, en los botones de los objetos de la tienda, el componente principal es una imagen con el icono de dicho accesorio, relegando a un segundo plano los textos en los que se especifica el nombre y tipo del objeto.

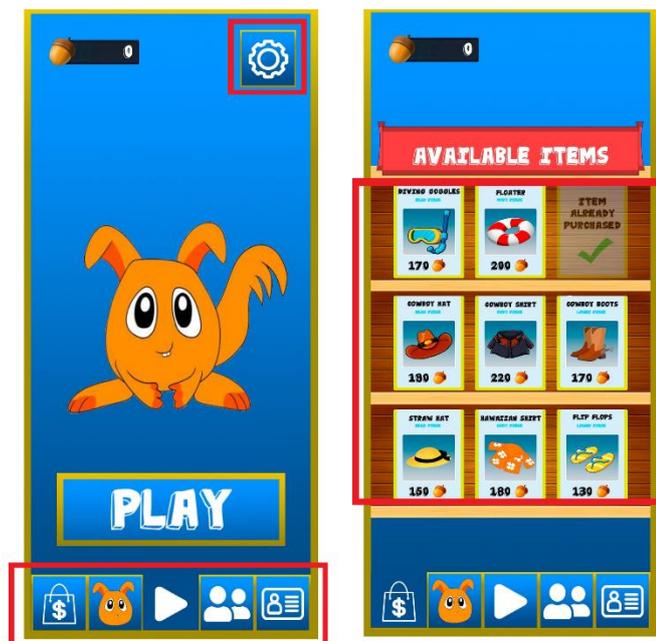


Figura 79. Botones con iconos

- **Es más fácil entender un diseño ordenado:** la estructura del videojuego se ha diseñado basándose en las leyes de Gestalt, que proporcionan una serie de teorías que permiten entender el mundo a partir de cómo están organizados sus elementos. Algunas de estas leyes se pueden ver en la Figura 80, que muestra la pestaña del perfil de usuario. Por una parte, los objetos cercanos forman grupos (en este caso minijuegos), siguiendo la ley de proximidad. Asimismo, se pueden apreciar claramente dos columnas; las partidas jugadas a la izquierda y las victorias a la derecha, siguiendo las leyes de continuidad y simetría.

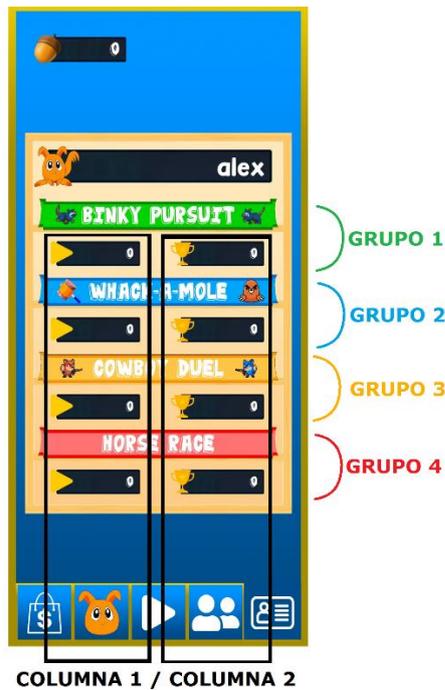


Figura 80. Leyes de Gestalt

### 7.3 Diagrama de flujo

Un diagrama de flujo es una representación gráfica que muestra la secuencia lógica o los pasos a seguir para navegar entre las diferentes pantallas de la aplicación. Estas se conectan mediante flechas, que indican que se puede acceder a la ventana destino desde la ventana origen. Este tipo de diagramas permite mostrar la totalidad de la aplicación y ayuda a entender cómo funciona y cuáles son las posibilidades que ofrece. Adicionalmente, presenta el estilo del producto en conjunto, ya que se pueden ver todas las ventanas y cómo se combinan entre sí.

A continuación, en la Figura 81, se muestra el diagrama de flujo del videojuego *Blitz Party*, señalando la navegación entre ventanas. Empezando desde arriba, la primera pantalla a la que se accede es el menú de registro. Una vez introducidos los datos del usuario, se llega a la ventana central del menú principal, desde la que se puede abrir el menú de ajustes o navegar a las otras cuatro ventanas que componen este menú. El menú de ajustes permite cerrar la cuenta y volver a la ventana de registro.

Asimismo, desde la ventana central se puede navegar a la pestaña de selección de partida (tanto multijugador como contra la IA). Desde este punto, se pasa a la pantalla de carga y a continuación, a los propios minijuegos. Finalmente, desde los minijuegos se abre el menú de pausa, que devuelve al usuario al menú principal.



Figura 81. Diagrama de flujo

## 8. Implementación del producto

---

### 8.1 Menús del juego

Las pantallas que componen la aplicación han sido implementadas siguiendo un mismo estilo que ayude al usuario a ubicarse y a aprender con rapidez qué es lo que representa y cuál es la funcionalidad de cada uno de los elementos que aparecen. El juego está formado por una ventana de registro, un menú principal dividido en 5 secciones y navegable gracias al grupo de botones situados en la parte inferior de la pantalla, un conjunto de menús sobre la pestaña actual (ajustes, menú de pausa y menú de selección de minijuegos) y los diferentes minijuegos disponibles. Estas pantallas se explican con más detalle a continuación:

- **Menú de registro e inicio de sesión** (Figura 82): Primera ventana que se muestra al iniciar la aplicación. Permite a los usuarios acceder a la aplicación mediante la introducción de un nombre de usuario y una contraseña, que se guardarán en la nube junto con todos los datos de dicho usuario. Si el nombre de usuario ya existe cuando se registra, se mostrará un mensaje de error indicando el cuál ha sido el fallo. Si por otra parte el nombre de usuario o la contraseña son erróneos cuando inicia sesión, se mostrará otro mensaje explicándolo. Si todo es correcto, entrará al juego.



Figura 82. Ventana de registro

- **Menú principal, ventana central** (Figura 83): Primera ventana que se muestra después de iniciar sesión. Muestra las bellotas ganadas en la esquina superior izquierda, nos permite acceder al menú de ajustes con el botón situado en la esquina superior derecha y desde ella accedemos al menú de selección de minijuego al pinchar en el botón central. La imagen central representa el personaje del jugador, un Zoomy.



Figura 83. Ventana central

- **Menú principal, tienda** (Figura 84): Ventana situada más a la izquierda, según el menú de botones inferior. Contiene los accesorios que se pueden equipar a nuestro personaje. Estos se organizan en grupos temáticos (ropa para la playa, indumentaria hawaiana, traje de vaquero, etc.) y se dividen en 3 tipos: cabeza, parte superior y zapatos. Para su adquisición, se debe pinchar en el botón correspondiente al objeto deseado y disponer de las bellotas necesarias, que se muestran con un número en la parte inferior de cada tarjeta. Al seleccionar la prenda, aparecerá una pestaña de confirmación que preguntará si de verdad se quiere comprar el objeto y, en caso afirmativo, se debe pulsar el botón de comprar que saldrá en dicha ventana.



Figura 84. Ventana de la tienda

- **Menú principal, inventario** (Figura 85): Esta ventana muestra todos los accesorios que el jugador ha comprado en la tienda, organizados en una lista deslizable, de modo que si no caben todos se puede navegar por la lista para encontrar la prenda buscada. Un clic en el objeto deseado abrirá un menú de confirmación que permite personalizar al Zoomy del usuario con dicho objeto, apareciendo en verde e indicando que está en uso. Al poder utilizar solo un accesorio de cada tipo a la vez, si se clican unos zapatos y ya había otros equipados, se preguntará si el usuario quiere sustituirlos. Finalmente, al pinchar sobre un objeto seleccionado, se podrá desequipar directamente.

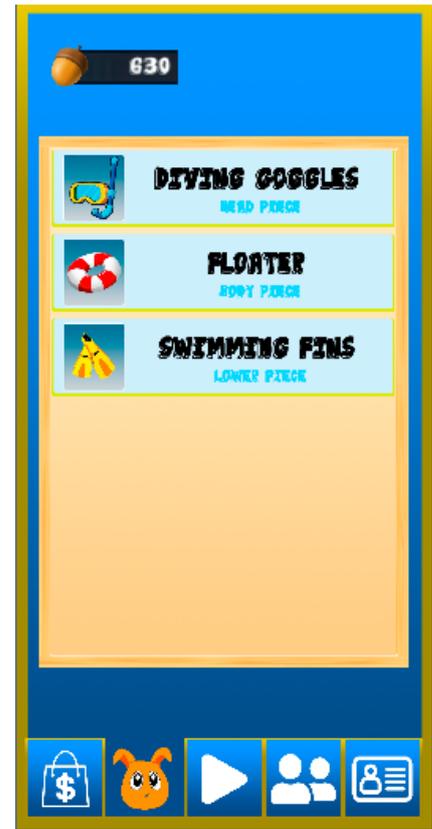


Figura 85. Ventana del inventario

- **Menú principal, perfil de usuario** (Figura 86): Se accede al pulsar el botón situado en la esquina inferior derecha. Contiene el nombre de usuario del jugador actual y las estadísticas de cada minijuego. A la izquierda, con el icono de *play*, se muestran las partidas jugadas y a la derecha, con el icono de una copa, las que se han ganado.

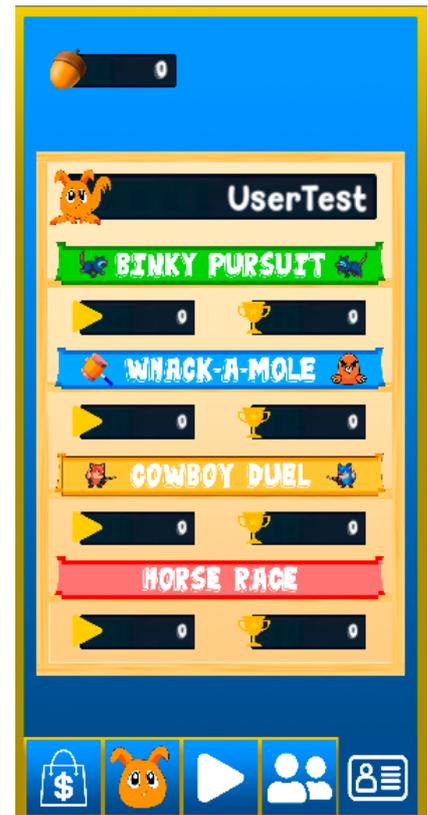


Figura 86. Ventana de perfil de usuario

- **Menú de selección de minijuego** (Figura 87): Permite escoger el minijuego que se desea jugar y se accede desde la ventana central del menú principal. Como los minijuegos están divididos por zonas y hay dos de cada una, se representan con colores diferentes siendo, por ejemplo, *Binky Pursuit* y *Whack-A-Mole* de la zona de prado y *Cowboy Duel* y *Horse Race* (aún por implementar) de la zona del desierto. Estos minijuegos se jugarán contra una IA y no en modo multijugador.

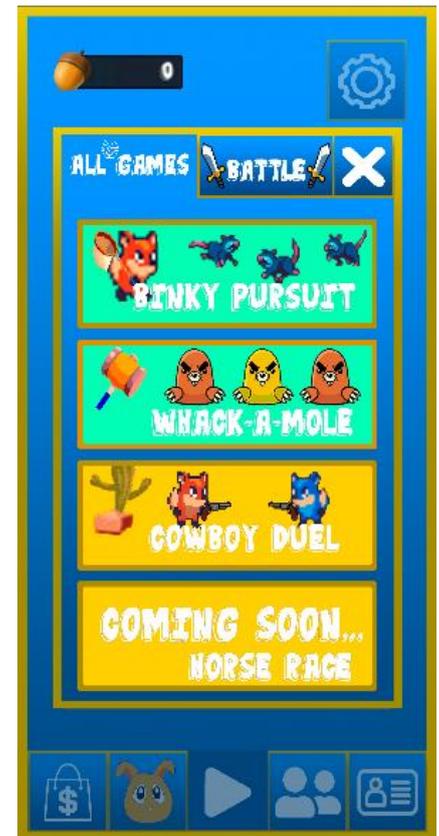


Figura 87. Menú selección de minijuego

- **Menú de modo batalla** (Figura 88): Podemos alternar entre este menú y el de selección de minijuego mediante los botones situados en la parte superior de este menú. Se ha introducido un párrafo que explica en qué consiste este modo: 3 minijuegos aleatorios que se jugarán en modo online contra otro usuario. Debajo aparece una animación con los dos personajes combatiendo y finalmente, el botón *play* que iniciará la batalla.

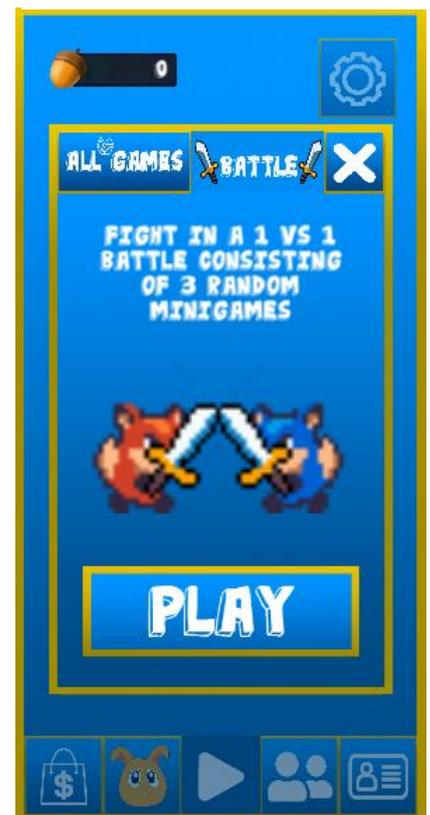


Figura 88. Menú modo batalla

- **Menú de ajustes** (Figura 89): Se puede acceder a él desde la ventana central del menú principal, mediante el botón situado en la esquina superior derecha. Ofrece la posibilidad de modificar el volumen del juego mediante dos barras deslizables que permiten ajustar tanto la música como los efectos de sonido. En el centro se muestra el personaje animado y en la parte inferior, el botón de cierre de sesión, que devuelve al usuario a la ventana de registro.

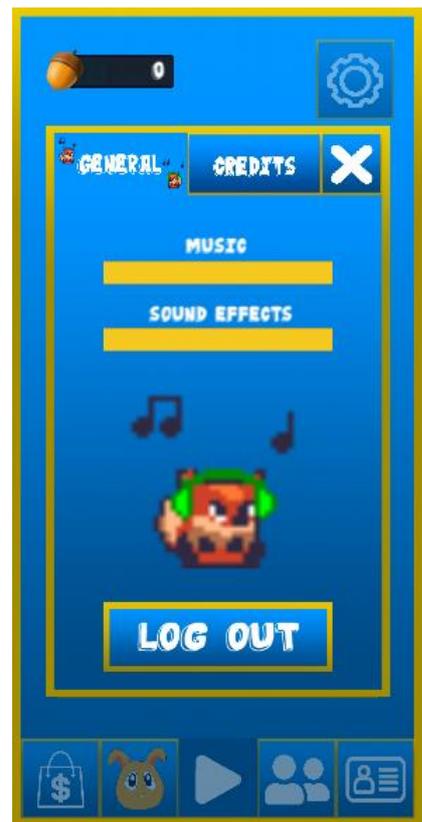


Figura 89. Menú de ajustes

- **Créditos** (Figura 90): La ventana de créditos se accede desde el menú de ajustes, mediante el botón situado en la parte superior. Lista el nombre del equipo, el de los desarrolladores del juego y las páginas web desde las que se han obtenido algunos de los recursos sonoros utilizados.



Figura 90. Ventana de créditos

- **Pantalla de carga** (Figura 91): Mostrada antes de que se inicie un minijuego. Inicia una animación divertida de los personajes persiguiéndose y presenta el minijuego que se va a cargar a continuación mediante una breve descripción sobre este.



Figura 91. Pantalla de carga

- **Minijuego *Binky Pursuit*** (Figura 92): El objetivo de este minijuego es conseguir atrapar más enemigos, *Binkies* en este caso, que el jugador rival. Para ello, se ha construido una interfaz que muestre las capturas de cada usuario en forma de puntuaciones, mostradas en las esquinas superiores. Además, un cronómetro situado en la parte superior indica el tiempo restante para que termine el minijuego y un botón en la esquina superior derecha permite pausar el juego y acceder al menú de pausa. Para poder mover al personaje y obtener puntos, se utiliza un *joystick* situado abajo a la izquierda y un botón de captura abajo a la derecha.



Figura 92. Minijuego Binky Pursuit

- **Minijuego *Whack-A-Mole*** (Figura 93): Para obtener más puntos que el rival en este minijuego, habrá que utilizar el martillo de que dispone el jugador para golpear a los topos que salen de los agujeros. Como en el minijuego anterior, hay un sistema de puntuaciones, cronómetro y pausa. Adicionalmente, cuando se clica la pantalla aparece el martillo para realizar un golpe. En caso de darle a un topo, este realizará una animación de muerte y se obtendrán puntos.



Figura 93. Minijuego Whack-A-Mole

- **Minijuego *Cowboy Duel*** (Figura 94): Simula los duelos de vaqueros del antiguo oeste. El menú de pausa y el sistema de puntuaciones se mantienen de los otros minijuegos. Los puntos se obtienen al salir victoriosos de una ronda y el que consigue dos, gana. Al principio de cada ronda se muestra una cuenta atrás y, cuando llega a cero, sale un cartel que urge a disparar. El primer jugador que pulse la pantalla después de este cartel, se lleva el punto.

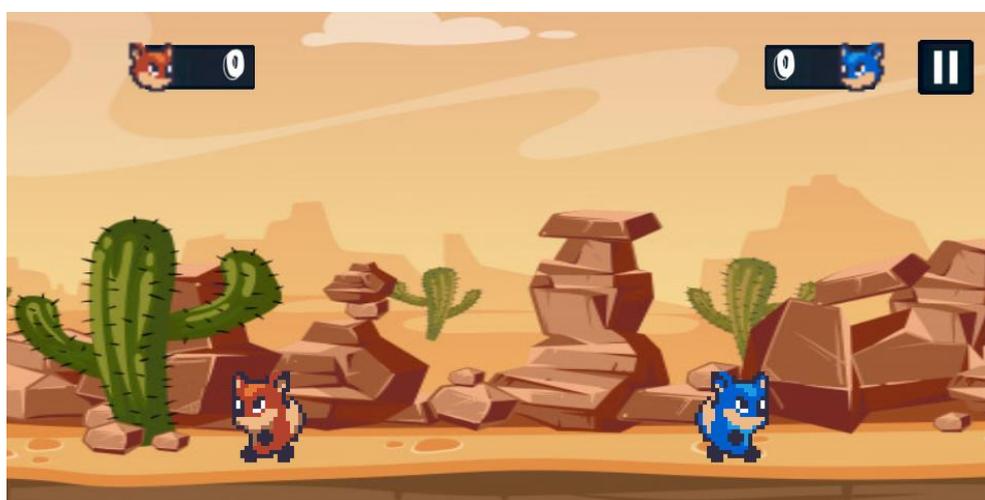


Figura 94. Minijuego Cowboy Duel

- **Menú de pausa** (Figura 95): Se muestra dentro de los minijuegos de un jugador cuando se pulsa el botón situado en la esquina superior derecha. Deja el juego de fondo pausado y ofrece la posibilidad de descansar y modificar el volumen del juego. Además, permite al jugador volver directamente al menú principal o reanudar la partida.



Figura 95. Menú de pausa

## 8.2 Funcionamiento de la tienda y el inventario

En este apartado se analizarán algunos *scripts* que son de especial importancia en lo que se refiere a la parte *front-end* del videojuego. A continuación, se explicará el funcionamiento de los métodos encargados de la compra de accesorios de la tienda y la administración del inventario.

### 8.2.1 *Script ShopManager*

En la Figura 96 se observan los métodos pertenecientes al *script ShopManager*, utilizados para la compra de objetos de la tienda. El método *ItemButtonHandler* se llama cuando se selecciona un objeto. Primero, se obtiene el coste en bellotas de dicho objeto. A continuación, se habilita el menú de confirmación de compra, que pregunta al jugador si está seguro de que quiere comprar el accesorio y actualiza el botón de comprar con el precio del objeto. Finalmente, hace una llamada al método *EnableButtons* para deshabilitar los botones de la tienda.

```

public class ShopManager : MonoBehaviour {
    public void ItemButtonHandler() {
        When object selected
    }
    private void EnableButtons(bool disabled) {
        Enables and disables buttons
    }
    public void PurchaseItemHandler() {
        When object purchased
    }
    private void UpdatePurchasedItem() {
        Called by PurchaseItemHandler
    }
}

```

Figura 96. Script *ShopManager*

El siguiente método, *PurchaseItemHandler*, se activa cuando el usuario clic en el botón de confirmación de compra. Comprueba si el jugador tiene bellotas suficientes para pagar el coste de la prenda seleccionada. En caso afirmativo, se procede a realizar la compra. Para ello, se activa el sonido de compra, se restan las bellotas gastadas por el usuario y se actualiza su valor en la interfaz. Posteriormente, se añade el accesorio al inventario y se llama al método *UpdatePurchasedItem* para actualizar la interfaz de la tienda, cambiando el botón del objeto adquirido por uno que indica que ya se ha realizado la compra. Por último, se cierra el menú de confirmación y se vuelven a activar los botones de la tienda. En caso de que el jugador no tenga suficientes bellotas para comprar, se mostrará un mensaje de error indicando el problema.

### 8.2.2 Script *InventoryManager*

Para mostrar al usuario los accesorios que ha adquirido en la tienda, se utiliza una lista deslizable en la pestaña del inventario. Esta lista puede enseñar a la vez un máximo de seis objetos y, cuando el jugador obtiene más de seis, permite que este pueda verlos todos deslizando la lista hacia arriba o hacia abajo con su dedo.

El método *UpdateItemList*, perteneciente al script *InventoryManager* y encargado del correcto funcionamiento de la lista de objetos adquiridos en la tienda. En caso de mostrar más de seis objetos, se calcula la altura que deberá tener la lista, cuyo valor se incrementará en un porcentaje por cada objeto extra. Además, se deberá desplazar la posición de la lista hacia abajo en el eje Y en función de los objetos extra que contenga. Esto se debe a que al agrandar la lista, se hace tanto hacia arriba como hacia abajo, por lo que se debe corregir su posición (*Figura 97*).



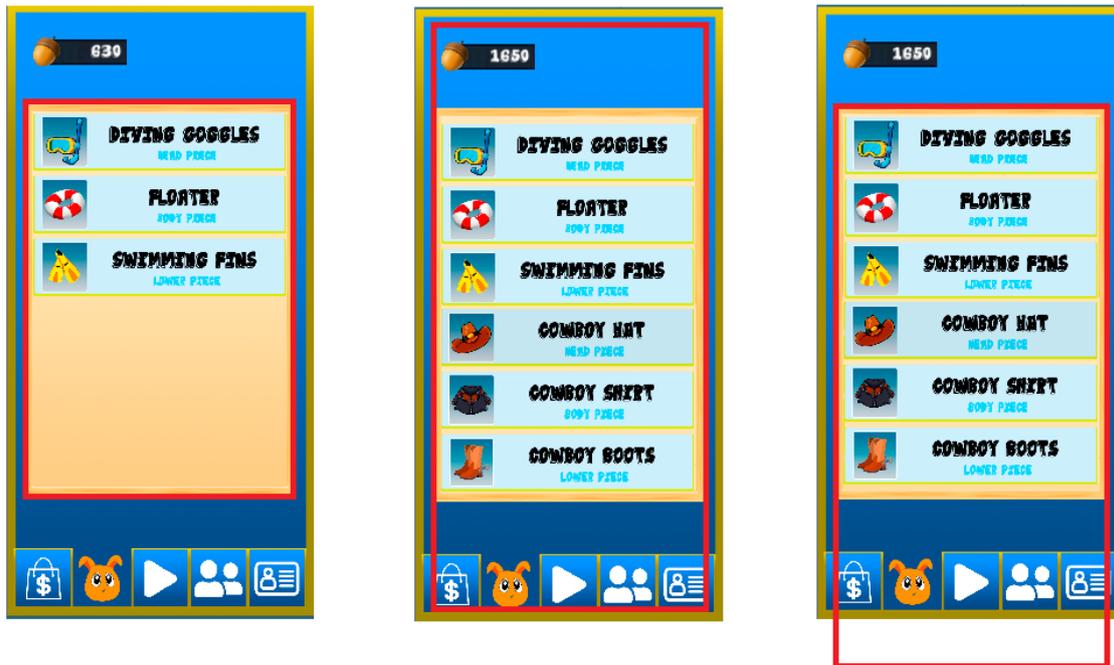


Figura 97. Funcionamiento de la lista deslizable

Seguidamente, se procede a introducir los objetos en la lista deslizable en forma de botones. Se instancian usando un *prefab* previamente creado, se les da el nombre y el *Sprite* correspondiente y se enlazan como hijos de la lista. Se les añade un método *listener* que se activará cuando se clique el botón y se calcula su posición dependiendo del orden de cada objeto en la lista.

Finalmente, y como se ha explicado en este punto, cuando se van a mostrar más de 6 objetos, la lista ha sido agrandada y su posición en el eje Y se ha tenido que modificar como se muestra en las dos últimas capturas de la *Figura 97*. Debido a esta diferencia de posición, el último paso consiste en desplazar los botones hacia arriba, pues la parte superior de la lista está ahora vacía, y los últimos botones no llegan a aparecer.

Con este método se consigue que la lista muestre todos los objetos adquiridos correctamente y que el deslizamiento sea fluido e intuitivo. Además, otorga al menú principal una característica muy utilizada en las aplicaciones para dispositivos móviles, la capacidad de deslizarse por la interfaz utilizando la pantalla táctil. Esta propiedad es muy útil, ya que la pantalla de los móviles es considerablemente más pequeña que la de un ordenador y por tanto la cantidad de información que puede exhibir es reducida.

### 8.3 Refactoring

Debido a la envergadura del proyecto, la cantidad de líneas de código y *scripts* programados es muy elevada. Por esta razón, es importante mantener una organización que permita entender el

código fácilmente e implementar cambios con rapidez. Con esta finalidad, se utiliza el *refactoring*.

El *refactoring* es el proceso de mejorar el código, sin crear ninguna funcionalidad nueva ni cambiar la ya existente, para producir un diseño más limpio y simple. Algunos de los métodos más utilizados son la extracción de código a otra clase y la extracción de código a otro método, y se explican a continuación.

Los métodos deben realizar solo una función concreta, que se ajuste al nombre que se le ha dado. Algunos de los que se habían implementado, en cambio, hacían más de una. Era el caso, por ejemplo, del método encargado de comprar objetos de la tienda: ejecutaba la acción de comprar, actualizaba el objeto de la interfaz que se ha comprado y habilitaba todos los botones cuando se cerraba el menú de compra.

Tras aplicar *refactoring* y como se puede observar en la Figura 98, estas tres funcionalidades se han dividido en tres métodos distintos, cada uno con un nombre significativo y un trabajo concreto: *PurchaseItemHandler* ahora solo realiza la compra, *UpdatePurchasedItem* refresca el objeto de la interfaz y *EnableButtons* activa todos los botones del inventario que se habían deshabilitado anteriormente.

```
public void PurchaseItemHandler()
{
    string itemName = selectedItem.name;
    int itemCost = Int32.Parse(selectedItem.GetComponentInChildren<Text>().text);
    errorText.SetActive(false);

    bool enoughAcorns = database.LoadAcorns() >= itemCost;

    if (enoughAcorns)
    {
        ServiceLocator.Instance.GetService<ISoundAdapter>().PlaySoundFX("BuyItemSFX"); //
        database.SaveAcorns(database.LoadAcorns() - itemCost); //Subtract acorns
        acornLabel.text = database.LoadAcorns().ToString(); //Update acorns label
        Item itemPurchased = new Item(itemName);
        database.AddItem(itemPurchased); //Add item to user's inventory in database
        UpdatePurchasedItem(); //Update item in the shop to purchased item
        confirmationMenu.SetActive(false); //Close menu
        EnableButtons(true); //Enable background buttons
    }
    else
    {
        errorText.SetActive(true);
    }
}

private void EnableButtons(bool disabled)
{
    item1.interactable = disabled;
    item2.interactable = disabled;
    item3.interactable = disabled;
    item4.interactable = disabled;
    item5.interactable = disabled;
    item6.interactable = disabled;
    item7.interactable = disabled;
    item8.interactable = disabled;
    item9.interactable = disabled;
    shopButton.interactable = disabled;
    zoomyButton.interactable = disabled;
    mainButton.interactable = disabled;
    friendsButton.interactable = disabled;
    profileButton.interactable = disabled;
}

private void UpdatePurchasedItem()
{
    selectedItem.GetComponentInChildren<Text>().text = "";
    selectedItem.GetComponent<Image>().sprite = Resources.Load<Sprite>("Item Purchased");
    selectedItem.GetComponent<Button>().interactable = false;
}
```

Figura 98. *PurchaseItemHandler* con *refactoring*

Por otro lado, si una clase contiene métodos que se refieren a una funcionalidad completa por sí sola y diferente a la de la clase que la contiene, se pueden extraer estos métodos a otra clase distinta, que englobe todas las acciones que tengan que ver con la funcionalidad extraída. Es el caso de la clase *PlayersScore*, que se encarga de controlar las puntuaciones de los jugadores.



En primera instancia, el control de las puntuaciones de los jugadores durante los minijuegos se llevaba a cabo desde una clase más extensa que administraba otras funciones del minijuego. Las puntuaciones, al ser un aspecto concreto y tener unos métodos específicos, se pueden extraer a una clase diferente, de forma que englobe dicho apartado.

De esta manera, todo el código relacionado con la gestión del contador de puntos y la actualización de las etiquetas de la interfaz que representan estos valores está localizado en la misma clase y es más sencillo de entender. Asimismo, facilita la escalabilidad de la aplicación, pues esta clase puede ser utilizada por otras introducidas más adelante en caso de que se decidieran crear más minijuegos.

# 9. Resultados obtenidos y pruebas

Al final del proceso de desarrollo, se debe comprobar que el producto implementado cumple con los requisitos especificados. Esto se verifica mediante la validación de requisitos, al final de cada *sprint* y utilizando la herramienta *Worki*. Asimismo, las encuestas preparadas para los experimentos de los MVP uno y dos permiten conocer la opinión de los usuarios y ayudan a tener una idea general sobre cuál ha sido el resultado final y cuál es la satisfacción de dichos usuarios.

## 9.1 Validación de requisitos

Para poder validar los requisitos establecidos al inicio del *sprint*, se especifican una serie de pruebas de aceptación (PA) para cada unidad de trabajo, con el objetivo de comprobar la verificabilidad de dichos requisitos. Por ejemplo, y como se puede observar en la Figura 99, las PA de la UT 2594 - Cowboy Duel: UI, Graphics and VFX son las siguientes:

- PA 5801 – La puntuación se actualiza correctamente cuando termina cada ronda.
- PA 5802 – La puntuación y las bellotas ganadas se muestran correctamente en el menú de resultados.
- PA 5803 – Se muestra el efecto de disparo cuando el usuario clicla la pantalla.

	Nombre PA	Última modificación de PA			
		Fecha modificación	UT	Sprint	Colaborador
	🔍	🔍	📅	🔍	🔍
⚡	5803 - Efecto Visual de Disparo	11/5/2021 19:28	2594	Sprint 3	
⚡	5802 - El resultado se muestra correctamente	11/5/2021 19:25	2594	Sprint 3	
⚡	5801 - La puntuación se actualiza correctamente	11/5/2021 19:24	2594	Sprint 3	

Figura 99. Pruebas de aceptación

Una vez se ha implementado la unidad de trabajo, y cuando se mueve a la columna de aplicar pruebas, se debe verificar que las funcionalidades desarrolladas corresponden a los requisitos especificados. Para ello, se probará manualmente cada una de las PA y se marcará si el comportamiento es el deseado o no, como aparece en la Figura 100. En caso de que se pasen satisfactoriamente todas las PA (Figura 101), la UT se terminará y, en caso contrario, se devolverá a la columna de programar, desde donde se habrán de arreglar los fallos encontrados.

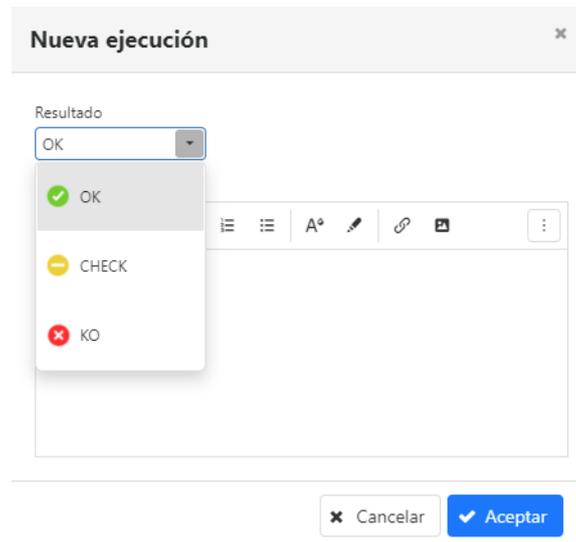


Figura 100. Ejecución de una PA

Nombre PA	Última modificación de PA				Última ejecución de PA				
	Fecha modificación	UT	Sprint	Colaborador	Resultado	Fecha ejecución	UT	Sprint	Colaborador
5803 - Efecto Visual de Disparo	11/5/2021 19:28	2594	Sprint 3		✓	11/5/2021 20:42	2594	Sprint 3	
5802 - El resultado se muestra correctamente	11/5/2021 19:25	2594	Sprint 3		✓	11/5/2021 19:56	2594	Sprint 3	
5801 - La puntuación se actualiza correctamente	11/5/2021 19:24	2594	Sprint 3		✓	11/5/2021 19:45	2594	Sprint 3	

Figura 101. PA ejecutadas satisfactoriamente

## 9.2 Encuesta primer MVP

Al final del tercer *sprint* y cuando se publicó el primer MVP en la *Play Store* de *Google*, se pidió a las personas que descargaron el juego que rellenasen una encuesta en la que se preguntaba sobre cuestiones relacionadas con diferentes aspectos de la aplicación. En lo que respecta al *front-end* del videojuego, su interfaz gráfica y sus menús, las dos preguntas que se incluyeron en la encuesta son las que se observan en la Figura 45 y en la Figura 46.

La primera pregunta tiene la finalidad de conocer si los usuarios consideran que la interfaz gráfica del videojuego es lo suficientemente clara. Un aspecto importante del diseño de los menús de la aplicación es que todas sus funcionalidades deben ser fáciles de entender y de aprender, no solo para los jugadores más veteranos, sino también para niños, adultos y gente más casual. Debido a que la encuesta se ha pasado a personas de todas las edades y las respuestas son positivas, se considera que lo que se ha implementado para el primer MVP es adecuado y no necesita cambios.

La segunda pregunta pretende saber la opinión de los usuarios respecto al diseño general del juego. En general, las puntuaciones obtenidas oscilan entre el siete y el diez, por lo que se entiende que el diseño de los personajes y de los minijuegos, así como el de los menús concuerdan con la ambientación y el estilo que se le pretende dar al videojuego.

Aunque los resultados obtenidos son muy positivos en general, no son perfectos. Por esta razón, se pensaron una serie de cambios menores que pretendían mejorar pequeños detalles de la interfaz y los menús:

- Se decidió cambiar el color de la mayoría de los textos, pues en su versión inicial eran de color amarillo y no se leían bien en algunos casos. El color decidido en su defecto fue el blanco, que sobre fondos azules y negros se entendía mucho mejor.
- Se arreglaron algunos *sprites*, cuya resolución no era buena y se veían difuminados. Los usuarios nos comentaron explícitamente que el zorro azul de Cowboy Duel no se veía bien, por lo que se solucionó este fallo.
- Se actualizaron los *sprites* de los objetos de la tienda, cambiando los colores para que se apreciara mejor su contenido, pues obtuvimos quejas de que no se diferenciaban bien.

### 9.3 Encuesta segundo MVP

En la encuesta realizada para el experimento del segundo MVP se han incluido varias preguntas para valorar la GUI. En la Figura 102 se puede observar que, en cuanto a si la interfaz resulta intuitiva a los usuarios, las respuestas obtenidas oscilan entre el seis y el diez, predominando las notas superiores al siete. Este resultado es importante, pues que los jugadores sean capaces de utilizar la interfaz de forma sencilla y clara es importante en este tipo de aplicaciones.

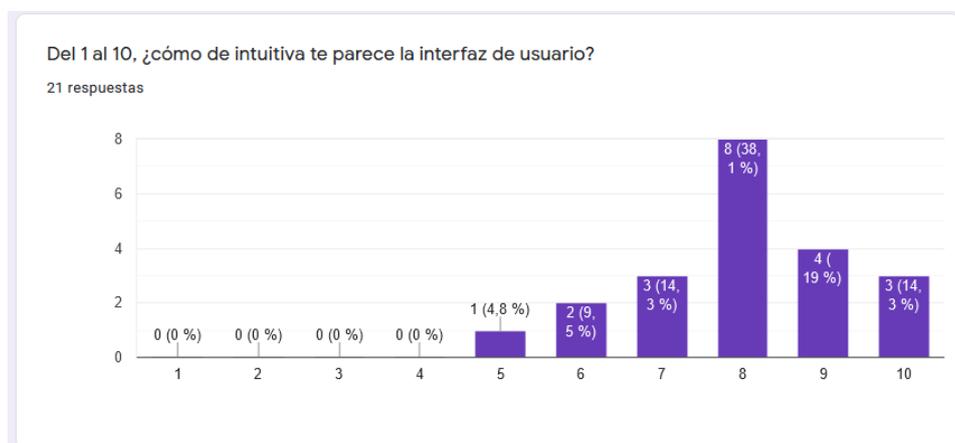


Figura 102. Pregunta MVP 2 interfaz intuitiva

Por otra parte, el atractivo de la interfaz suele ser uno de los puntos fuertes de muchos videojuegos, y la razón de que los usuarios sigan jugando. Para saber si el equipo de desarrollo

iba por el buen camino, se preguntó a los encuestados sobre este tema y el resultado fue muy positivo (Figura 103).

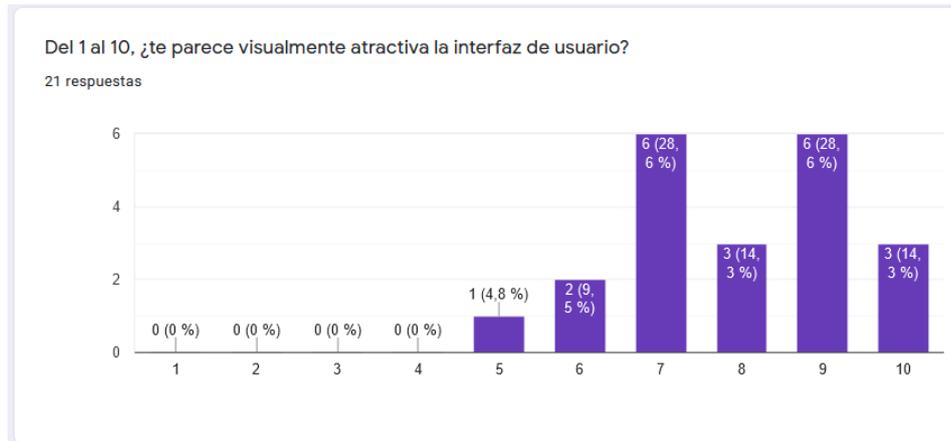


Figura 103. Pregunta MVP 2 interfaz atractiva

Asimismo, se ha preguntado también por el feedback que los jugadores reciben de la aplicación a las acciones que los usuarios realizan. La aplicación debería responder con algún tipo de estímulo (como efectos visuales o sonoros), de forma que los usuarios entiendan qué es lo que tienen que hacer. Además, en caso de realizar alguna acción errónea, la aplicación debería informar al jugador del fallo cometido. Este comportamiento se implementa en *Blitz Party*, por ejemplo, al indicar mediante una flecha qué personaje corresponde a cada jugador, o impidiendo al usuario comprar los accesorios que no puede pagar (Figura 104). Podemos comprobar, mediante los resultados de la pregunta mostrada en la Figura 105, que los encuestados están contentos con el *feedback* del juego.

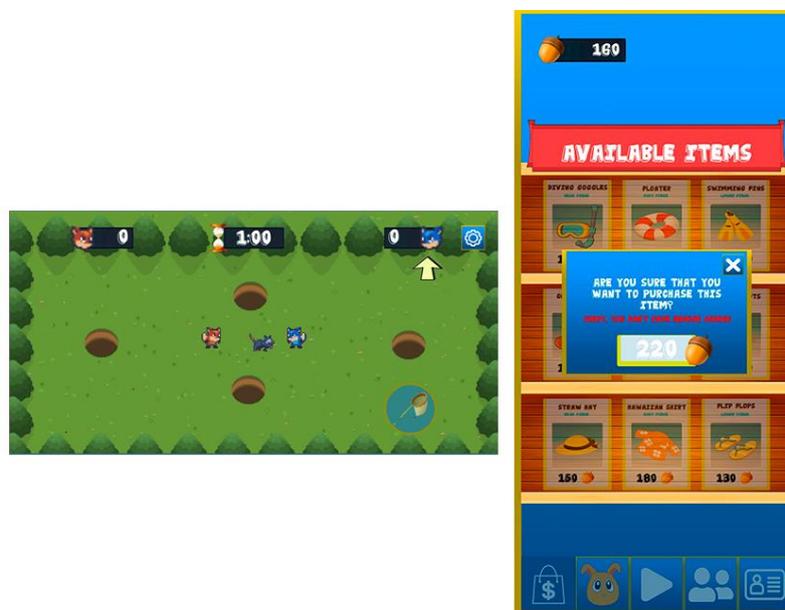


Figura 104. Feedback de la interfaz gráfica

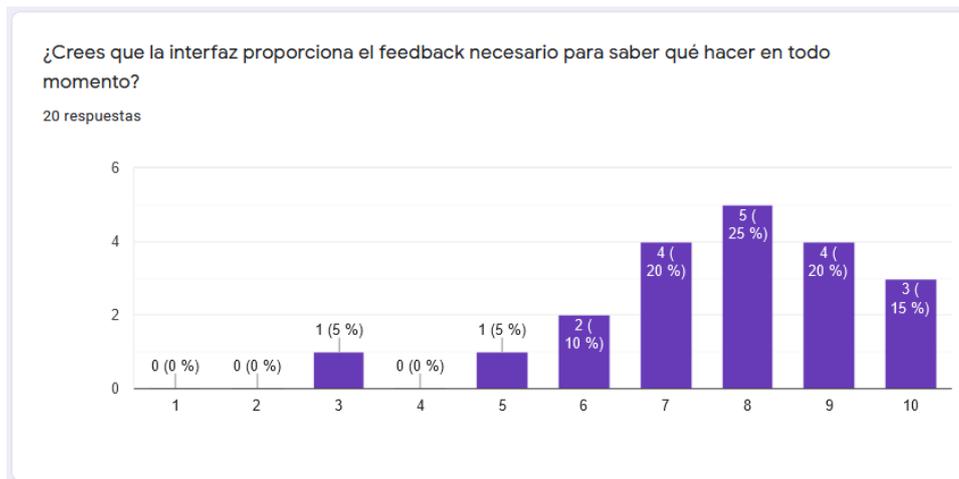


Figura 105. Pregunta MVP 2 *feedback* de la interfaz

Finalmente, se incluyó una pregunta de respuesta abierta cuyo objetivo era preguntar a los encuestados si añadirían alguna funcionalidad adicional a la interfaz y los menús del juego que no estuviese ya presente. Algunas de las respuestas han sido interesantes, como la posibilidad de desplazarse entre las pestañas del menú principal deslizando con el dedo. Otras, como la capacidad de interactuar con el personaje en la ventana central, ya se habían pensado. No obstante, realizar este tipo de encuestas otorga al equipo la capacidad de conocer los gustos y preferencias de los usuarios, y preparar el próximo MVP en consecuencia.

En general, los resultados obtenidos en la encuesta del segundo MVP han sido mejores que los de la primera, ya que siguiendo los consejos que se obtuvieron en la primera encuesta y como se ha explicado al final del epígrafe 9.2, se han realizado los cambios pertinentes para mejorar la aplicación. Sin embargo, hay numerosos cambios que aún pueden implementarse para aumentar la satisfacción de los jugadores, pues los resultados no han sido perfectos.

Respecto a la sencillez de uso, se pretende sustituir algunos textos por iconos, para que la interfaz sea más intuitiva, sobre todo para la gente mayor y los más pequeños. Estas imágenes, más fáciles de entender, serán más adecuadas que los textos actuales, muchas veces demasiado pequeños para que se puedan leer en dispositivos móviles.

En cuanto al *feedback* de la aplicación, se plantea añadir más formas de avisar al jugador de las acciones que está realizando. Por una parte, se añadirá una animación de captura en *Binky Pursuit* cuando se presione el botón de captura, se modificarán las alertas de la cuenta atrás de *Cowboy Duel* y se añadirán tutoriales al inicio de los minijuegos. Asimismo, se pretende cambiar el sistema de aviso de la tienda cuando no se tiene dinero suficiente para comprar algo. En lugar de permitir al jugador clicar el botón de comprar, se le mostrará directamente el mensaje de error, evitando así el fallo del usuario.

## 10. Conclusiones

---

La realización de este trabajo ha permitido cumplir satisfactoriamente los objetivos establecidos en el epígrafe 1.3 por los motivos explicados a continuación. El primer objetivo consistía en la realización de dos MVP, con sus experimentos correspondientes. A lo largo del proceso de desarrollo se han producido estos dos MVP, y se han llevado a cabo experimentos con los jugadores, realizando encuestas y obteniendo *feedback*. Por tanto, se puede afirmar que se ha cumplido este objetivo.

El segundo objetivo se centraba en participar en un proyecto de emprendimiento de principio a fin, desde la concepción de la idea de negocio hasta la salida del producto al mercado, y en la utilización de metodologías ágiles. Durante la implementación del videojuego, se han podido aplicar los conocimientos adquiridos en diferentes asignaturas de la carrera. Las metodologías ágiles como método de organización del equipo en un entorno de emprendimiento, aprendido en PSW (Proceso de Software) y PIN (Proceso de Ingeniería de Software), han facilitado el trabajo en equipo y han asegurado la obtención de resultados. Esto nos ha permitido cumplir el segundo objetivo.

Asimismo, como resultado del proyecto, se ha podido generar una aplicación completa y publicarla en la *Play Store* de *Android*, cumpliendo así con el tercer objetivo mencionado, que exigía subir el producto final a esta plataforma. Además, se ha implementado una interfaz atractiva, adaptada para dispositivos móviles y sencilla de entender y aprender. Todo esto se ha hecho sin disponer de ningún diseñador gráfico que nos ayudase con los modelos. Como resultado, se ha cumplido el objetivo número cuatro, que requería que la interfaz fuese a la vez funcional y sencilla de utilizar.

Los objetivos quinto y sexto requerían que el diseño del *front-end* se llevase a cabo siguiendo las fases de análisis, diseño, implementación y pruebas; y además el diseño de la interfaz tenía que seguir los principios de diseño de interfaces gráficas. Como se ha explicado en los capítulos del seis al nueve, se han analizado los requisitos, se ha diseñado una interfaz siguiendo los principios mencionados, se ha implementado la solución y se han realizado pruebas con usuarios, cumpliendo así con estos dos objetivos.

Por último, pero no menos importante, cabe destacar el trabajo realizado no solo como equipo de desarrollo, sino también como miembro de ese equipo. La sociedad ha cambiado mucho en las últimas décadas, y con ella la concepción que se tiene sobre cómo debería desempeñarse un

empleo. Trabajar solo o aislado de los demás profesionales no es una opción viable hoy en día, y mucho menos en un marco de emprendimiento y como ingeniero informático.

Por ello y gracias a haber efectuado este proyecto con los compañeros de equipo, cuya aportación y cooperación han sido fundamentales, se ha vivido una oportunidad enriquecedora que ha permitido experimentar en profundidad cuál debe ser el papel de un desarrollador. No una persona que trabaja sola delante de la pantalla, abstraída de los demás deberes como profesional, sino alguien capaz de trabajar en equipo, entender a los usuarios y superarse continuamente para producir la mejor solución posible a los problemas planteados.



# 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).

UT: Unidad de Trabajo.

PA: Prueba de Aceptación.

## 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.....	103
Versión 1.00 .....	103
Versión 2.00 .....	103
Versión 3.00 .....	103
2. Visión global del juego .....	103
Filosofía .....	103
Preguntas frecuentes.....	103
3. Conjunto de características.....	104
Características generales.....	104
Características multijugador .....	104
Editor.....	105
Jugabilidad .....	105
4. El mapa del mundo.....	105
Visión general .....	105
Zonas temáticas .....	105
Parque en constante evolución.....	106
Sistema de renderizado .....	106
Cámara .....	106
Motor de videojuegos.....	107
5. Personajes del juego .....	107
Visión general .....	107
Creando un personaje .....	108
Personajes especiales para cada minijuego .....	108
6. Interfaz de usuario .....	108
Visión general .....	108
7. Herramientas .....	111
Visión general .....	111
Variedad de herramientas .....	111
8. Banda sonora y efectos de sonido.....	112

Visión general .....	112
Sonido 3D.....	112
9. Minijuegos .....	112
<i>Binky Pursuit</i> .....	113
<i>Whack-A-Mole</i> .....	116
<i>Cowboy Duel</i> .....	118
10. Juego para un jugador .....	121
Visión general .....	121
Historia.....	121
11. Juego multijugador.....	121
Visión general .....	121
Batalla de minijuegos .....	121
Máximo de jugadores .....	122
Servidores.....	122
Personalización.....	121
Internet .....	122
Persistencia.....	122
Guardado y cargado.....	122
12. Calendario de trabajo .....	123



# 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 106 podemos ver los bocetos originales de los *Zoomies* junto a un simple modelo.



Figura 106. 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 107 se observa el diseño inicial del menú principal del videojuego.



Figura 107. 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 108). 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.

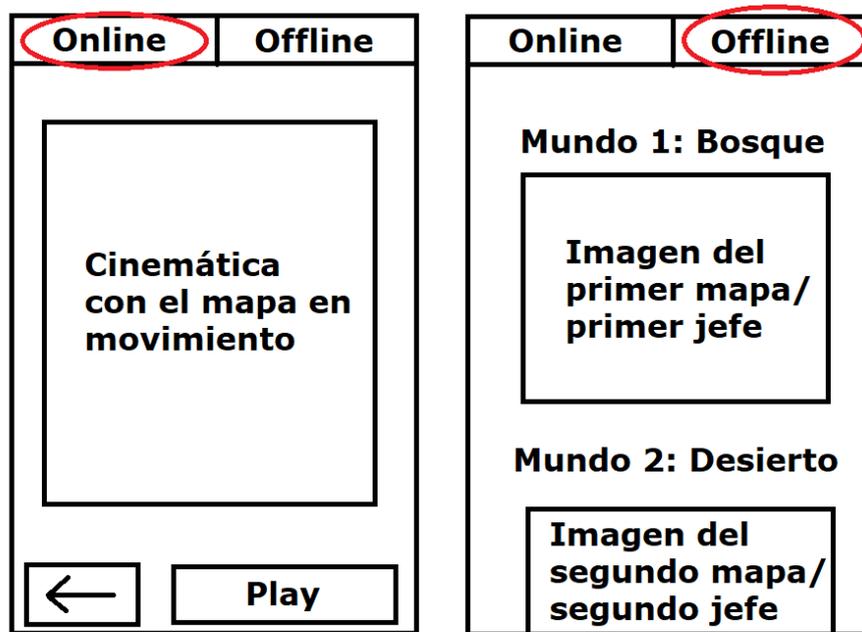


Figura 108. Pantallas de juego

## 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 109 se pueden observar ejemplos de herramientas que se utilizarán en estos minijuegos.



Figura 109. 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 110) 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.

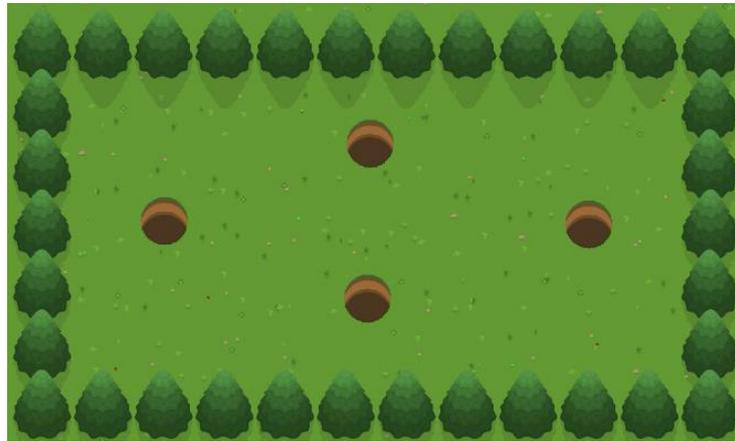


Figura 110. Escenario de *Binky Pursuit*

### Interfaz de usuario

Como se aprecia en la Figura 111, 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.

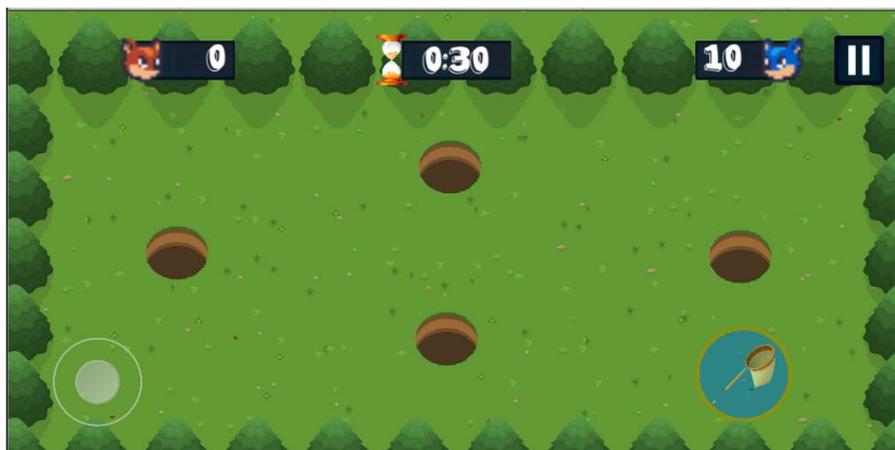


Figura 111. Interfaz de *Binky Pursuit*.

## Personajes

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

- Zorros (Figura 112): Los personajes controlables por los jugadores.



Figura 112. Zorros de *Binky Pursuit*

- *Binkies* (Figura 113): 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 113. *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 114) simula una llanura donde hay siete madrigueras de topos. En los bordes del mapa existen vallas que delimitan el mapa.

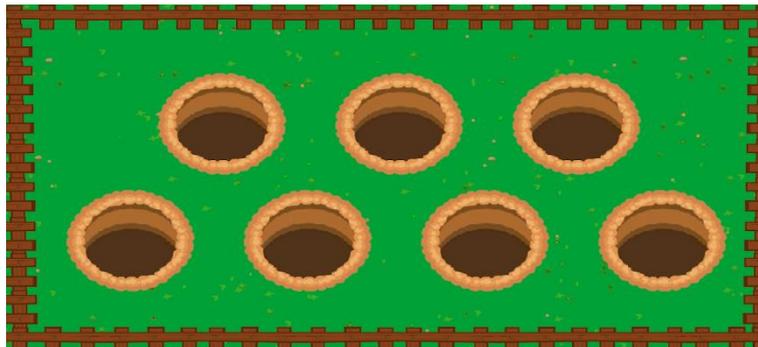


Figura 114. Escenario de *Whack-A-Mole*

### **Interfaz de usuario**

Como se puede ver en la Figura 115, 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 115. 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 116): es el personaje que controla el jugador. Simula el martillo real del arcade original *Whack-A-Mole* (Guacamole en España).



Figura 116. Martillo de *Whack-A-Mole*

- Topo (Figura 117): es el enemigo estándar. Golpearle con el martillo otorga un punto.



Figura 117. Topo normal de *Whack-A-Mole*

- Topo dorado (Figura 118): es un topo especial que tiene menor probabilidad de aparecer que el topo. Golpearle con el martillo otorga cinco puntos.



Figura 118. Topo dorado de *Whack-A-Mole*

- *Zoomy* (Figura 119): es el personaje del jugador. Golpearle con el martillo resta tres puntos.



Figura 119. *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 120) simula un desierto del lejano oeste donde los vaqueros solían tener este tipo de duelos.



Figura 120. Escenario de *Cowboy Duel*

### Interfaz de usuario

Como se puede apreciar en la Figura 121, 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 121. Interfaz de *Cowboy Duel*

### Personajes

En este minijuego participan los dos zorros, uno de cada jugador (Figura 122).



Figura 122. 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 123. 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>>

