



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Desarrollo de un videojuego en Unity: implementación de las mecánicas de juego

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Jose Francisco Gómez Alemany

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 de desarrollo de sus TFG, donde cada uno se ha centrado en un área del proyecto.

Este trabajo se enfoca en la implementación de las diversas mecánicas de juego que alberga el producto. Las mecánicas son todos los elementos y sistemas que gobiernan el videojuego y dotan a este de una buena jugabilidad. Estos factores se han ideado para que los jugadores tengan la mejor experiencia posible al interactuar con el videojuego. Para crear estas mecánicas y optimizar el desarrollo del proyecto, se han usado un abanico de técnicas y se han analizado los resultados.

Palabras clave: videojuego, Unity, minijuegos, emprendimiento, mecánicas, jugabilidad.

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 una àrea del projecte.

Aquest treball es centra en l'implementació de les diferents mecàniques de joc de les què està format el producte. Les mecàniques són tots els elements i sistemes que governen el videojoc i li proporcionen una bona jugabilitat. Aquests factors s'han ideat per a que els jugadors tinguin la millor experiència possible quan interacciona amb el videojoc. Per a crear aquestes mecàniques i optimitzar el desenvolupament del projecte, s'han utilitzat un conjunt de tècniques i s'han analitzat els resultats.

Paraules clau: videojoc, Unity, minijocs, empenedoria, mecàniques, jugabilitat.

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 area of the project.

The focus of this work is the implementation of the various game mechanics that form the product. The mechanics are all the elements and systems that rule over the video game and provide it with quality gameplay. These aspects have been devised to give the player the best possible experience when interacting with the video game. To create this mechanics and optimize the development, a range of techniques have been used and the results have been analyzed.

Keywords: video game, Unity, minigames, entrepreneurial, mechanics, gameplay.

Tabla de contenidos

1. Introducción.....	12
1.1 Motivación	13
1.2 Justificación del tema	14
1.3 Objetivos	14
1.4 Estructura de la memoria.....	15
2. Evaluación de la idea de negocio	16
2.1 Estado del arte	16
2.2 Estudio de mercado	19
2.2.1 Super Mario Party	20
2.2.2 Wario Ware	22
2.2.3 Among Us	24
2.2.4 Dumb Ways to Die	26
2.3 Análisis DAFO.....	28
2.4 Modelo de negocio y proyección económica.....	29
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>	52
3.3.1 Desarrollo del <i>MVP</i>	53
3.3.2 Experimento	54
3.4 Despliegue del producto y <i>marketing</i>	57
4. Tecnologías utilizadas	59
4.1 Unity Engine	59



4.2 Visual Studio	59
4.3 Git/GitHub	60
4.4 Worki Process	61
4.5 Photoshop.....	61
4.6 Mirror	62
4.7 Azure PlayFab.....	62
5. Análisis del problema.....	64
5.1 Elicitación de requisitos	64
5.1.1 Brainstorming	64
5.1.2 Entrevistas	65
5.2 Definición de requisitos.....	66
5.2.1 Requisitos funcionales	67
5.2.2 Requisitos no funcionales	68
5.3 Especificación de requisitos	68
6. Diseño de la solución	70
6.1 Introducción	70
6.2 Binky Pursuit.....	71
6.2.1 Diseño de la inteligencia artificial	71
6.2.2 Movimiento del Jugador.....	72
6.2.3 Captura de Binkies	72
6.3 Whack-A-Mole	73
6.3.1 Movimiento y golpeo	73
6.4 Cowboy Duel	74
6.4.1 Disparo	74
7. Unity3D	75
7.1 Interfaz de Usuario	75
7.2 Escenas.....	77
7.3 GameObjects y Prefabs	78
7.4 Componentes.....	79

7.5 Scripts	81
8. Implementación.....	83
8.1 Patrón Object Pool	83
8.2 Modularidad.....	84
8.1 Binky Pursuit.....	85
8.1.1 Movimiento del Jugador.....	85
8.1.2 Generador de Binkies	86
8.1.3 Captura de Binkies	87
8.2 Whack-A-Mole	88
8.2.1 Movimiento del mazo.....	88
8.2.2 Generador de enemigos	89
8.2.3 Golpeo de enemigos	89
8.3 Cowboy Duel	90
8.3.1 Disparo	90
9. Pruebas y resultados obtenidos.....	92
9.1 Validación de requisitos	92
9.2 Encuestas a usuarios	93
10. Conclusiones	96
11. Trabajo futuro	97
12. Apéndices.....	98
12.1 Listado de abreviaturas, siglas y acrónimos.....	98
12.2 Lista de recursos gráficos	99
12.3 <i>Game Design Document</i> (GDD).....	100
13. Referencias bibliográficas	127

Listado de figuras

Figura 1. Logo de Blitz Party	12
Figura 2. Enlace QR de descarga de Blitz Party	13
Figura 3. Lanzamiento de jabalina en Olympic Decathlon	16
Figura 4. Partida de Bomberman Party Edition	17
Figura 5. Personaje personalizable de la Nintendo Wii	18
Figura 6. Captura del videojuego Buzz!: el Gran Reto	18
Figura 7. Cronología juegos party o de minijuegos	19
Figura 8. Tabla comparativa de productos competidores	20
Figura 9. Portada de Super Mario Party	21
Figura 10. Modo tablero de Super Mario Party	21
Figura 11. Minijuego de Super Mario Party	22
Figura 12. Portada de Wario Ware Gold	23
Figura 13. Microjuego de Wario Ware Gold	23
Figura 14. Cinemática del modo historia de Wario Ware Gold	24
Figura 15. Portada de Among Us	25
Figura 16. Impostor asesinando a un tripulante en Among Us	25
Figura 17. Jugador realizando una tarea en Among Us	26
Figura 18. Portada de Dumb Ways to Die	27
Figura 19. Fotograma de la canción Dumb Ways to Die	27
Figura 20. Microjuego de Dumb Ways to Die	28
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	35
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	45
Figura 37. Edad de los participantes de la encuesta.....	46
Figura 38. Tiempo que dedican a videojuegos los participantes de la encuesta.....	46
Figura 39. Porcentaje de participantes que juegan en dispositivos móviles.....	46
Figura 40. Horas que juegan los participantes de la encuesta en dispositivos móviles	47
Figura 41. Minijuego favorito de los encuestados	47
Figura 42. Puntuaciones sobre la dificultad del juego	48
Figura 43. Puntuaciones de la curva de aprendizaje	48
Figura 44. Opinión sobre dificultad excesiva de algún minijuego	49
Figura 45. Opiniones sobre la claridad de la interfaz gráfica.....	49
Figura 46. Opiniones sobre diseño y ambientación	50
Figura 47. Opinión sobre la música de los minijuegos	50
Figura 48. Puntuaciones generales de Blitz Party	51
Figura 49. Mapa de características del segundo MVP	52
Figura 50. Binky Pursuit multijugador	53
Figura 51. Whack-A-Mole multijugador.....	54
Figura 52. Pregunta rango de edad	55
Figura 53. Pregunta jugadores de dispositivos móviles	55
Figura 54. Pregunta horas jugadas en dispositivos móviles	56
Figura 55. Pregunta minijuego preferido	56
Figura 56. Pregunta puntuación general Blitz Party	57
Figura 57. Blitz Party en la Play Store	57
Figura 58. Países desde los que se ha descargado Blitz Party	58
Figura 59. Cuenta de Twitter de Swifter Games.....	58
Figura 60. Logo de Unity Engine	59
Figura 61. Logo del IDE Visual Studio	59
Figura 62. Logos de Git y GitHub respectivamente	60
Figura 63. Logo de Worki TUNE-UP Process	61
Figura 64. Logo de Adobe Photoshop	61
Figura 65. Logo de Mirror.....	62
Figura 66. Logo de Azure PlayFab.....	63
Figura 67. Ideas brainstorming.....	65
Figura 68. Pregunta controles Binky Pursuit	66
Figura 69. Salto en el videojuego Super Mario Bros	70
Figura 70. Captura mostrando la ruta calculada por el algoritmo pathfinding	71
Figura 71. Máquina de estados de la IA del rival en Binky Pursuit	72

Figura 72. Captura del videojuego Scott Pilgrim vs. The World	72
Figura 73. Captura del videojuego Fruit Ninja	74
Figura 74. Quick time event del videojuego Uncharted 3	74
Figura 75. Editor de Unity.....	75
Figura 76. Escenas de Blitz Party	77
Figura 77. Listado de GameObjects ofrecidos por Unity	78
Figura 78. Prefabs utilizados en Blitz Party (iconos en azul)	79
Figura 79. Vista del componente Transform desde el inspector de Unity	79
Figura 80. Vista del componente Rigidbody 2D desde el inspector de Unity	80
Figura 81. Sprite de un enemigo con el collider visible.....	80
Figura 82. Ejemplo visual del patrón object pool obtenida de raywenderlich.com.....	83
Figura 83. Interfaz del cronómetro	84
Figura 84. Máquina de estados del personaje en Binky Pursuit.....	85
Figura 85. Diagrama de flujo de la función Move()	86
Figura 86. Captura de Binky Pursuit con los enemigos visibles activos en la Object Pool.....	86
Figura 87. Diagrama de flujo del algoritmo responsable de la generación de Binkies.....	87
Figura 88. Diagrama de flujo de la función PerformAction()	88
Figura 89. Enemigos del minijuego Whack-A-Mole.....	89
Figura 90. Máquina de estados del topo	90
Figura 91. Máquina de estados personaje Cowboy Duel	91
Figura 92. Pruebas de aceptación del minijuego Whack-A-Mole.....	92
Figura 93. Fases de una unidad de trabajo.....	93
Figura 94. Pregunta dificultad general segundo MVP	94
Figura 95. Pregunta dificultad de minijuegos segundo MVP	94
Figura 96. Pregunta sobre la facilidad de control	95
Figura 97. Personajes de Blitz Party	109
Figura 98. Mockup menú principal Blitz Party	111
Figura 99. Pantallas de juego	112
Figura 100. Herramientas de los minijuegos	113
Figura 101. Escenario de Binky Pursuit	114
Figura 102. Interfaz de Binky Pursuit.	115
Figura 103. Zorros de Binky Pursuit	115
Figura 104. Binkies de Binky Pursuit.....	115
Figura 105. Escenario de Whack-a-Mole	117
Figura 106. Interfaz de Whack-a-Mole	117
Figura 107. Martillo de Whack-a-Mole.....	118
Figura 108. Topo normal de Whack-a-Mole	118

Figura 109. Topo dorado de Whack-a-Mole	118
Figura 110. Zoomy de Whack-a-Mole	118
Figura 111. Escenario de Cowboy Duel.....	120
Figura 112. Interfaz de Cowboy Duel	121
Figura 113. Personajes de Cowboy Duel.....	121
Figura 114. Calendario de trabajo	126

1. Introducción

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

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



Figura 1. Logo de *Blitz Party*

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



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

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

- Alejandro Vicent Micó: desarrollo del apartado *front-end* del videojuego.
- Jose Francisco Gómez Alemany: desarrollo de las mecánicas del videojuego, descrito en esta memoria.
- 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 e implementar unas mecánicas de juego que resulten satisfactorias para el jugador.
- Crear una base sólida utilizando elementos modulares y reutilizables para facilitar la implementación de futuros minijuegos.

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 desarrolla la parte individual del proyecto, que en este caso comienza con el análisis del problema que se quiere solventar con este trabajo. En el siguiente capítulo se presenta el diseño de la solución seguido de un apartado de introducción al motor *Unity3D* útil para comprender el siguiente capítulo sobre la implementación de las mecánicas de juego. Finalmente, se expone una sección donde se validan los requisitos y se comprueban los resultados obtenidos.

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



2. Evaluación de la idea de negocio

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

2.1 Estado del arte

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

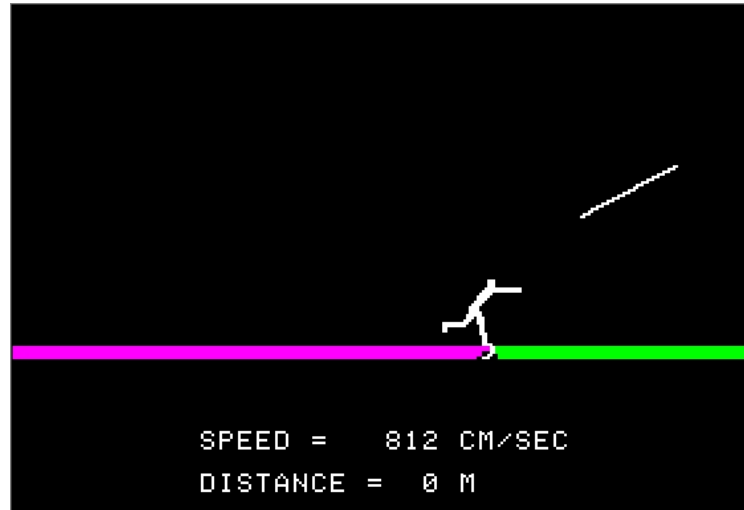


Figura 3. Lanzamiento de jabalina en *Olympic Decathlon*

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

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

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

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



Figura 4. Partida de *Bomberman Party Edition*

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

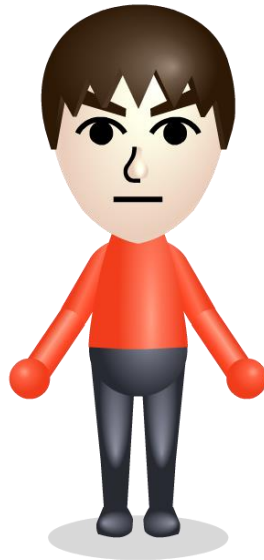


Figura 5. Personaje personalizable de la *Nintendo Wii*

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



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

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

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

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



Figura 7. Cronología juegos *party* o de minijuegos

2.2 Estudio de mercado

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

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

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

Comparativa Características de productos competidores				
Nombre	Super Mario Party	WarioWare Gold	Among Us	Dumb Ways to Die
Desarrollador	NDeube	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*

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



Figura 10. Modo tablero de *Super Mario Party*

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

hacer rebajas con sus juegos estimamos que ha recaudado aproximadamente 450 millones de euros.

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



Figura 11. Minijuego de *Super Mario Party*

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

Otro punto negativo de *Super Mario Party* es su elevado precio. *Super Mario Party* tiene actualmente un precio de mercado de 59,95€. El público objetivo de este tipo de videojuegos consiste en jugadores que buscan jugar partidas rápidas y de bajo compromiso, por lo que el elevado despliegue económico que conlleva la compra de este juego es un gran inconveniente.

2.2.2 Wario Ware

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



Figura 12. Portada de *Wario Ware Gold*

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

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



Figura 13. Microjuego de *Wario Ware Gold*

Las fortalezas competitivas de *Wario Ware Gold* son la corta duración de sus microjuegos, así como el estilo desenfadado que tiene el juego. Ambas características se aprecian en la Figura 13.

Debido a que se puede jugar una partida de manera rápida y que además el juego está disponible en una plataforma portátil como es la *Nintendo 3DS*, los jugadores pueden jugarlo en cualquier momento. Además, el estilo gráfico de los microjuegos es variable y se adapta a cada uno de ellos. Finalmente, el modo historia (Figura 14) tiene una naturaleza de comedia, lo que alimenta ese estilo desenfadado.



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

Como debilidades competitivas destacan la falta de multijugador tanto local como online. Esto puede provocar que los microjuegos, pese a ser variados, se acaben volviendo repetitivos dado que al no haber un contrincante todas las partidas son iguales. La segunda debilidad competitiva que encontramos en *Wario Ware Gold* es, al igual que en el caso de *Super Mario Party*, su elevado precio. Pese a que es más barato (29,95€), consideramos que sigue siendo un precio demasiado alto para un juego de este estilo.

2.2.3 Among Us

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



Figura 15. Portada de *Among Us*

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

El principal motivo del triunfo de este juego es su componente social. Si un jugador encuentra un cadáver en algún punto de la nave puede reportarlo y eso producirá una reunión en la que todos los jugadores tienen que discutir e intentar descubrir a los impostores o, en el caso de estos impostores, defenderse y convencer al resto de jugadores de que son inocentes.



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

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

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



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

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

2.2.4 Dumb Ways to Die

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

de descargas y la campaña publicitaria de la que forma parte ha conseguido un beneficio de unos 6 millones de dólares en impresiones de los medios [5].

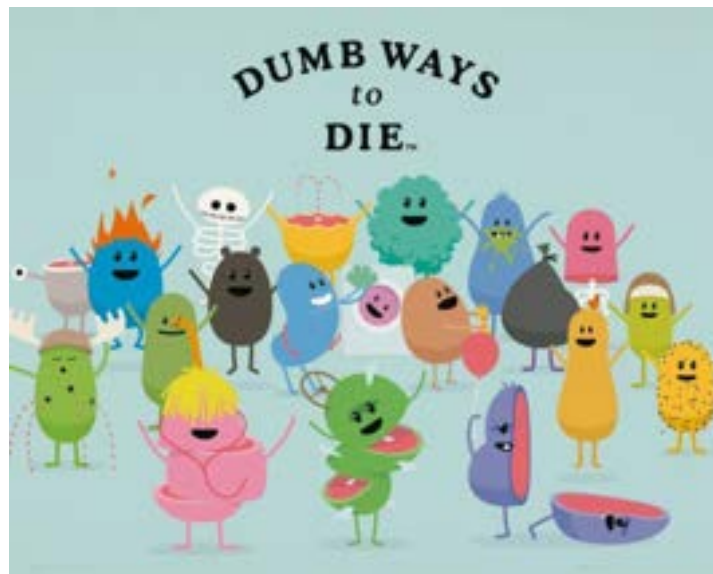


Figura 18. Portada de *Dumb Ways to Die*

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

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



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

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

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 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
Total de dispositivos que tienen el juego	20000	19000	17000
Ingresos Trimestrales			
Media anuncios vistos * ganancia por anuncio	1,500 €	1,425 €	1,275 €
Media micropagos hechos * ganancia media por micropago	1,000€	950€	850€
Total ingresos	2,500 €	2,375 €	2,125 €
Gastos Trimestrales			
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 €
Total Gastos	40,660 €	35,660 €	35,660 €
Resultado Trimestral	-38,160 €	-33,285 €	-33,535 €
Resultado Trimestral Acumulado	-38,160 €	-71,445 €	-104,980 €

Figura 22. Proyección económica año 1

A lo largo del primer año se espera una decaída en el número de descargas y jugadores, por lo que el primer trimestre del segundo año se procede a aumentar el presupuesto destinado al marketing, con la finalidad de reflotar el juego. A partir de este punto, se espera que 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 de las mecánicas de juego

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
Total empleados	5	5	5	5	5	7	9	9	9	9

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
Total de dispositivos que tienen el juego	1480000	1220000	980000

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.).

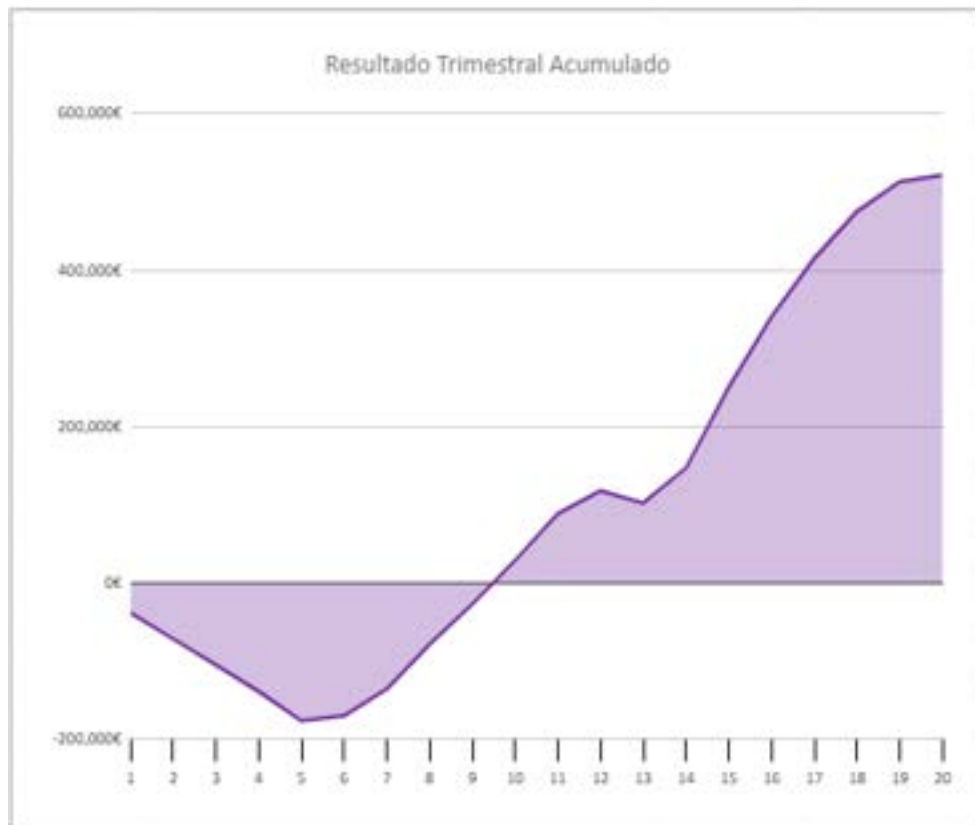


Figura 27. Resultado acumulado de ingresos

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

7 Costos

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

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

6 Ingresos

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

· Primer año:

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

· Segundo año:

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

Figura 28. Lean Canvas de *Blitz Party*

2.6 Conclusiones

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

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

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

3. Desarrollo de la idea de negocio

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

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

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

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

Además, nos reunimos con Patricio Letelier, profesor de las asignaturas Proceso de Software (PSW) y Proyecto de Ingeniería de Software (PIN). Nos permitió utilizar la herramienta Worki para organizar el trabajo en equipo utilizando metodologías ágiles y nos dio 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.

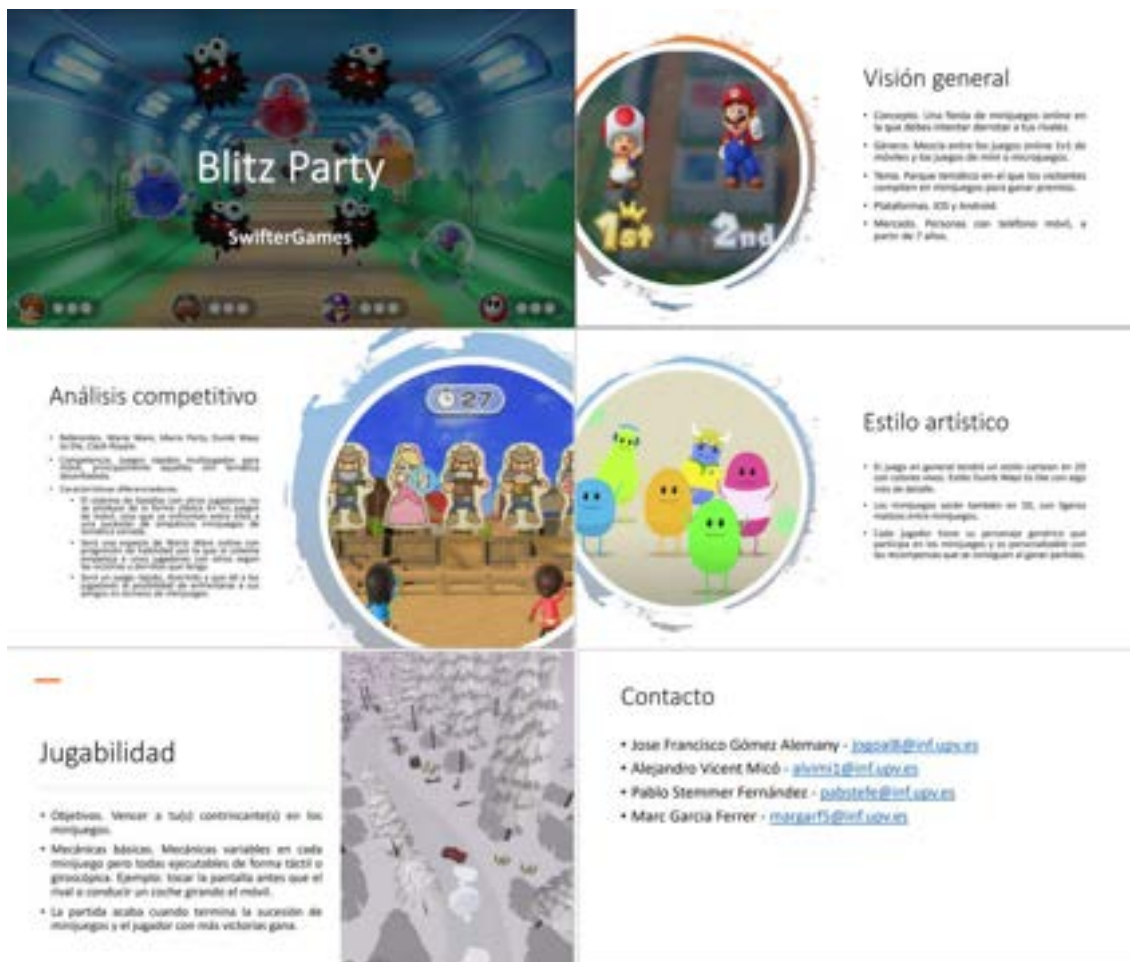


Figura 29. *Pitch Doc*

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 alce victorioso en dos rondas obtendrá la victoria.



Figura 35. Captura del minijuego *Cowboy Duel*

3.2.2 Experimento

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



Figura 36. Tuit de promoción del primer MVP

Con este 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 le gustaban a los jugadores. A continuación, se muestran las respuestas obtenidas en el experimento y se detalla cómo se han usado para hacer cambios e incluir mejoras.

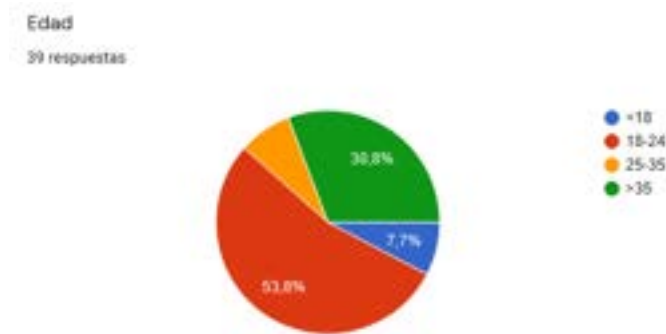


Figura 37. Edad de los participantes de la encuesta

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

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

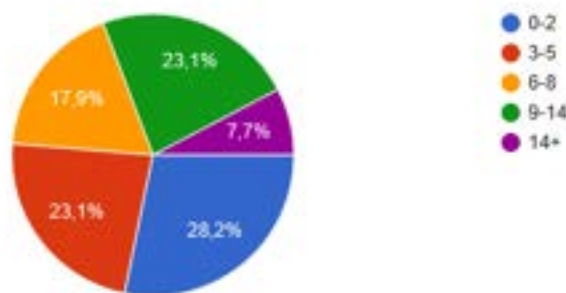


Figura 38. Tiempo que dedican a videojuegos los participantes de la encuesta

¿Sueles jugar a videojuegos en dispositivos móviles?
39 respuestas

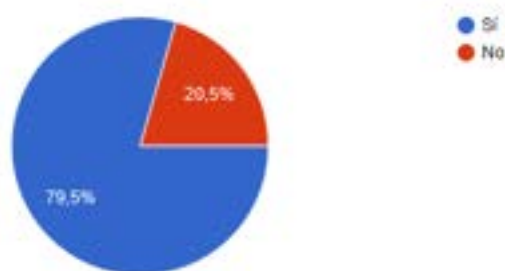


Figura 39. Porcentaje de participantes que juegan en dispositivos móviles

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

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

33 respuestas

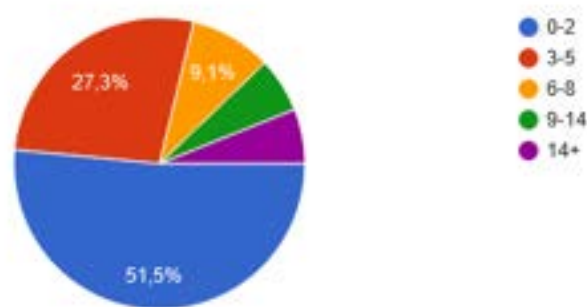


Figura 40. Horas que juegan los participantes de la encuesta en dispositivos móviles

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

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

39 respuestas

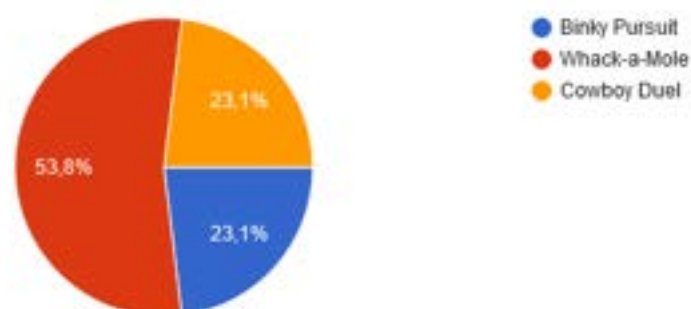


Figura 41. Minijuego favorito de los encuestados

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

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

39 respuestas

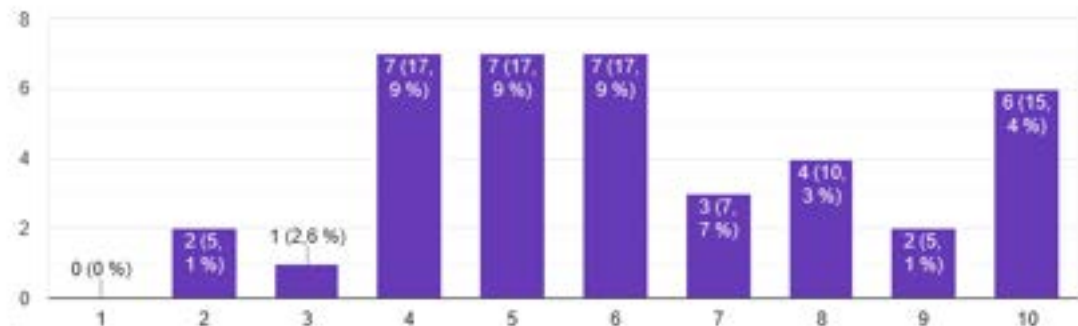


Figura 42. Puntuaciones sobre la dificultad del juego

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

39 respuestas

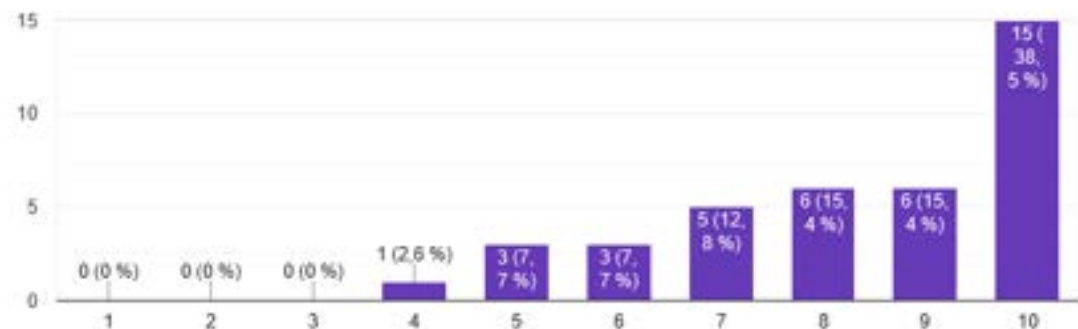


Figura 43. Puntuaciones de la curva de aprendizaje

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.



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.



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.



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.



Figura 47. Opinión sobre la música de los minijuegos

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

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

39 respuestas

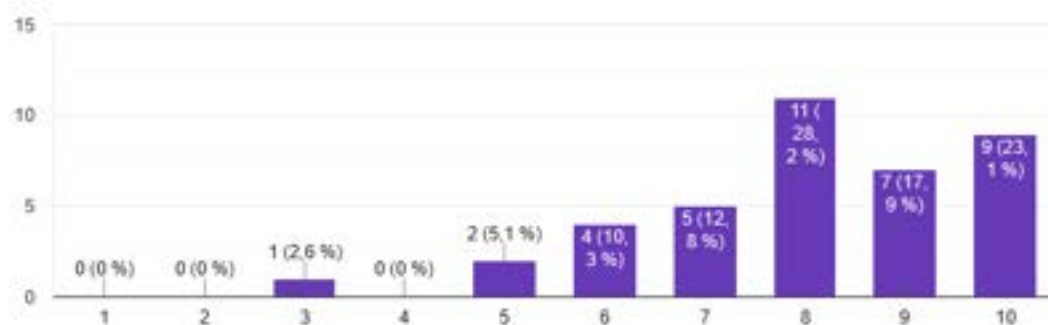


Figura 48. Puntuaciones generales de Blitz Party

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

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

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

3.3 Segundo *Minimum Viable Product*

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



Figura 49. Mapa de características del segundo MVP

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

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

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

3.3.1 Desarrollo del MVP

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

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

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



Figura 50. *Binky Pursuit* multijugador

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

distinguir a ambos jugadores. El funcionamiento del minijuego no ha cambiado, se deben golpear a los topes 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.

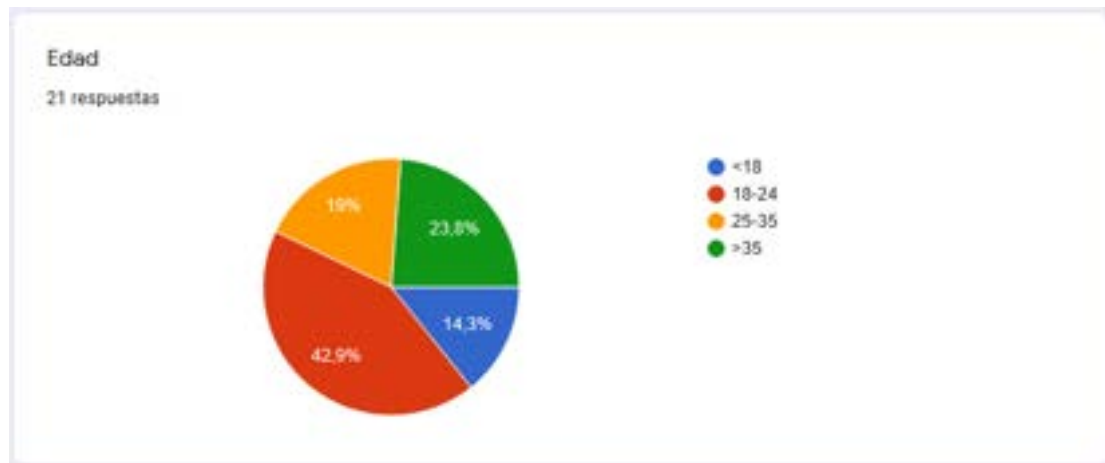


Figura 52. Pregunta rango de edad

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.

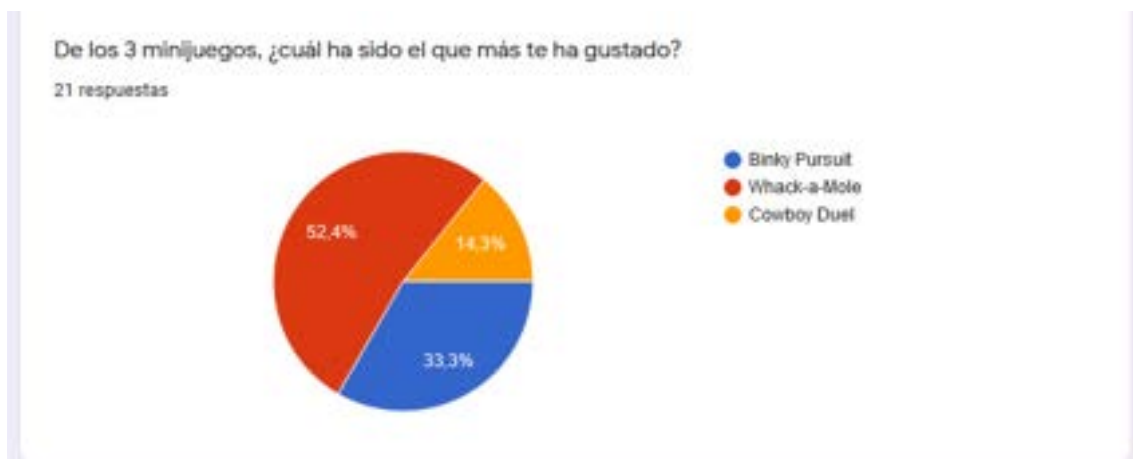


Figura 55. Pregunta minijuego preferido

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.

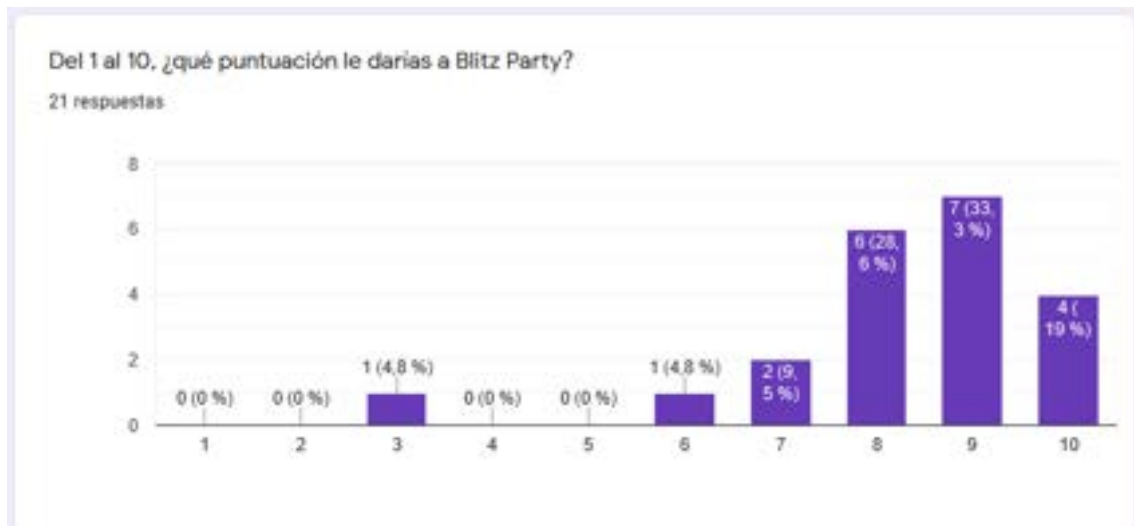


Figura 56. Pregunta puntuación general Blitz Party

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 ve en la Figura 58.

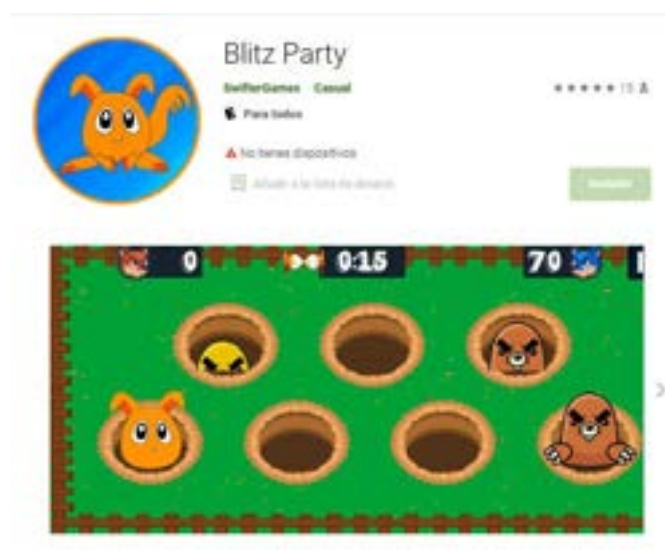


Figura 57. *Blitz Party* en la *Play Store*

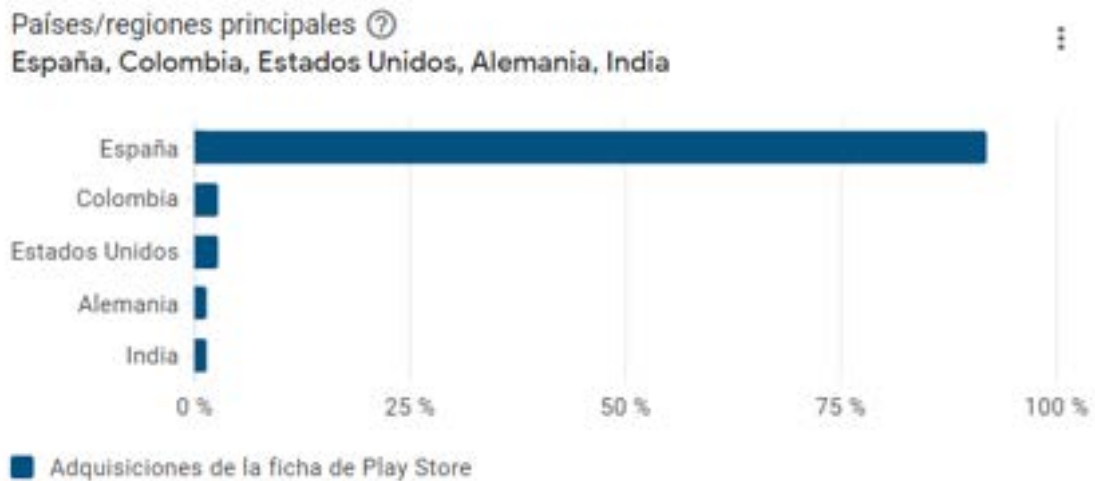


Figura 58. Países desde los que se ha descargado *Blitz Party*

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



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

4. Tecnologías utilizadas

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

4.1 Unity Engine



Figura 60. Logo de *Unity Engine*

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

4.2 Visual Studio



Figura 61. Logo del IDE *Visual Studio*

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

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

4.3 Git/GitHub



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

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

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

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

4.4 Worki Process



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

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

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

4.5 Photoshop



Figura 64. Logo de *Adobe Photoshop*

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

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

4.6 Mirror

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



Figura 65. Logo de *Mirror*

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

4.7 Azure PlayFab

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



Figura 66. Logo de *Azure PlayFab*

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

5. Análisis del problema

Este proyecto se ha considerado como un proyecto software completo y por ello ha sido necesario pasar por el proceso de Ingeniería de Requisitos (IDR) visto en la asignatura AER (Análisis y Especificación de Requisitos) de la rama de Ingeniería del Software. En este proceso entran en juego la identificación, definición y especificación de los requisitos para más tarde poder validarlos. Es vital que el producto final que se desea crear cumpla con unos requisitos que se plantean con anterioridad al comienzo del desarrollo. Es importante recordar que, al seguir una metodología ágil, muchos de los requisitos que se muestran a continuación sufrieron cambios durante el desarrollo del videojuego.

En este tipo de proyectos, usualmente existe una relación entre desarrollador y cliente que tiene el objetivo de facilitar la identificación de los requisitos para crear una aplicación conforme a los estándares originales del cliente. Sin embargo, en este caso al ser un proyecto totalmente independiente, los propios miembros del equipo han desempeñado ambos roles.

5.1 Elicitación de requisitos

En esta primera fase se tiene como objetivo entender los objetivos del proyecto e identificar los requisitos que la aplicación debe cumplir para alcanzar los objetivos. En este caso se han utilizado dos técnicas bastante comunes: *brainstorming* y entrevistas.

5.1.1 Brainstorming

Esta técnica consiste en la recolección de ideas e información de todos los *stakeholders* y se divide en tres fases: preparación, generación y consolidación. Para realizar esta técnica los cuatro miembros del equipo se reunieron de manera telemática y utilizaron una pizarra virtual para apuntar las ideas. En primer lugar, se determinaron las metas que se deseaban alcanzar con este proyecto. En este caso, nos interesaba crear un videojuego para dispositivos móviles con diversos minijuegos que se pudieran jugar tanto en solitario como en modo multijugador.

Seguidamente, nos adentramos en la segunda fase en la que se debían generar todas las ideas posibles relacionadas con los objetivos que se concluyeron en la fase anterior. Cabe recalcar que en esta fase es importante no discriminar ideas que pueden resultar poco convencionales ya que el propio objetivo de esta fase es el de generar tantas ideas como sean posibles para, en la tercera y última fase, poder organizarlas y clarificarlas. Se puede observar el resultado de esta lluvia de ideas en la Figura 67 donde, además, se han marcado las relacionadas con el tema de este trabajo: las mecánicas y la jugabilidad.

cambios que consideramos que han sido positivos para crear un mejor producto. Podemos ver un ejemplo de esto en la Figura 68.

En este caso se preguntaba acerca de los controles del minijuego *Binky Pursuit*. Debido a que es un juego para dispositivos móviles los controles deben hacer uso de la pantalla táctil de estos dispositivos, pero existen varias maneras de implementarlos. En este minijuego se debe controlar el movimiento de un personaje en la pantalla lo cual puede resultar algo tosco sin la ayuda de periféricos externos como mandos con *joysticks*. Por este motivo, se buscó que los controles fueran lo más simples y cómodos para el jugador y se pensaron dos opciones viables: un sistema donde el personaje se mueve hacia los toques de pantalla del jugador y la incorporación de un *joystick* virtual donde el jugador tiene control absoluto del movimiento del personaje.

Para la primera opción, se usaría el algoritmo *pathfinding* para que el personaje del jugador tuviera la habilidad de esquivar obstáculos, en este caso los agujeros de las madrigueras. Se creó una pequeña demo en la que simplemente se podía controlar un personaje con estos controles para que los entrevistados pudieran tener una toma de contacto. Después de estudiar los resultados de la entrevista, se ha decidido utilizar el *joystick* debido a resultar más satisfactorio y práctico tener el control total del personaje. De todos modos, se ha reutilizado el sistema de *pathfinding* para la IA de los *Binkies* y el rival como se explica en los próximos apartados.

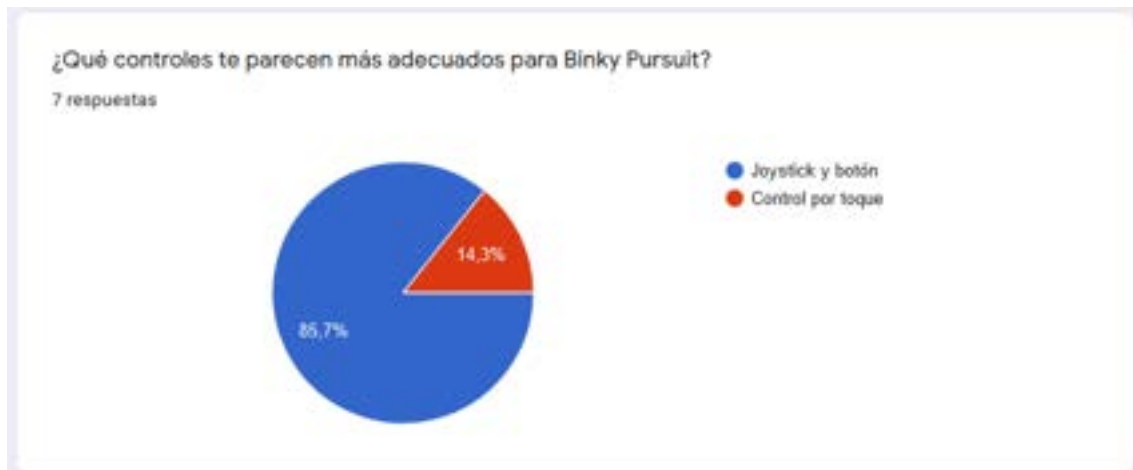


Figura 68. Pregunta controles *Binky Pursuit*

5.2 Definición de requisitos

Una vez finalizada la elicitación de requisitos, se pasa a la fase de definición de requisitos. Aquí se describirán las funcionalidades del sistema y las características del funcionamiento del videojuego. Para ello, se han listado tanto los requisitos funcionales; que describen las funcionalidades que ofrece el sistema, como los no funcionales; referentes a las restricciones de estas funcionalidades, relacionados con las mecánicas y la jugabilidad.

5.2.1 Requisitos funcionales

Nombre	Elegir modo de juego
Descripción	El juego permitirá al usuario elegir entre dos modos de juego: juego libre y modo batalla.

Nombre	Elegir minijuego
Descripción	En caso de haber elegido juego libre, el usuario podrá elegir el minijuego al que quiere jugar.

Nombre	Elegir multijugador o en solitario
Descripción	El usuario podrá elegir si desea jugar en solitario contra una IA o enfrentarse a otro rival en modo multijugador.

Nombre	Pausar minijuego
Descripción	El usuario podrá pausar los minijuegos siempre que esté jugando en solitario.

Nombre	Controlar personaje en <i>Binky Pursuit</i>
Descripción	El usuario podrá manualmente controlar a su personaje en el minijuego <i>Binky Pursuit</i> .

Nombre	Capturar <i>Binky</i>
Descripción	En el minijuego <i>Binky Pursuit</i> el usuario deberá poder capturar a los <i>Binkies</i> cuando esté encima de ellos y se presione el botón correspondiente.

Nombre	Golpear enemigo
Descripción	En el minijuego <i>Whack-A-Mole</i> el usuario deberá poder golpear a los enemigos que aparecen de las madrigueras.

Nombre	Disparar
Descripción	En el minijuego <i>Cowboy Duel</i> el usuario realizará un disparo si realiza la acción correspondiente una vez aparezca la señal que permite disparar.

Nombre	Obtener puntuación
Descripción	En todos los minijuegos el usuario deberá poder obtener puntos al realizar las acciones correspondientes de cada minijuego: <ul style="list-style-type: none">• <i>Binky Pursuit</i>: capturar un <i>Binky</i>.• <i>Whack-A-Mole</i>: golpear a un enemigo.• <i>Cowboy Duel</i>: disparar antes que el rival.

Nombre	Recompensa al finalizar minijuego
Descripción	El usuario obtendrá <i>acorns</i> al acabar el minijuego en caso de haber empatado o ganado.

5.2.2 Requisitos no funcionales

- El juego deberá funcionar en dispositivos móviles con el sistema operativo Android.
- El juego deberá adaptarse a dispositivos de diferentes resoluciones y tamaños.
- El tiempo de carga no deberá superar los tres segundos.
- El juego no recopilará ningún tipo de datos de los usuarios.
- Se realizarán guardados constantes de los *acorns* de los usuarios para poder recuperarlos en caso de error del sistema.

5.3 Especificación de requisitos

Finalmente se llega a la fase de especificación de requisitos donde se detalla y precisa toda la información conseguida. El propósito de esta fase es detallar qué se requiere de los desarrolladores y así tener la funcionalidad concreta que se desea implementar. Para realizar esta especificación se han usado las tablas de los casos de uso y se centrará en los requisitos de las mecánicas de los minijuegos.

Nombre	<i>Binky Pursuit – Mover personaje</i>
Descripción	El usuario es capaz de mover a su personaje.
Actores	Usuario
Precondiciones	El usuario está dentro del minijuego <i>Binky Pursuit</i> .
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario toca el lado izquierdo de la pantalla para hacer aparecer un <i>joystick</i>. 2. El usuario desliza el joystick en la dirección deseada. 3. El personaje se mueve en la dirección del <i>joystick</i>.

Nombre	<i>Binky Pursuit – Capturar Binky</i>
Descripción	El usuario es capaz de capturar <i>Binkies</i> .
Actores	Usuario
Precondiciones	El usuario está dentro del minijuego <i>Binky Pursuit</i> .
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario mueve el personaje hasta estar encima de un <i>Binky</i>. 2. El usuario presiona el botón de captura. 3. El <i>Binky</i> desaparece y el jugador ha obtenido un punto.

Nombre	<i>Whack-A-Mole – Golpear enemigo</i>
Descripción	El usuario es capaz de golpear a los enemigos.
Actores	Usuario
Precondiciones	El usuario está dentro del minijuego <i>Whack-A-Mole</i> .
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario toca a un enemigo.

	<ol style="list-style-type: none"> 2. Aparece el martillo del jugador y el enemigo realiza una animación. 3. El jugador obtiene un punto al golpear a un topo, tres puntos al golpear a un topo dorado y pierde un punto al golpear a un <i>Zoomy</i>.
--	--

Nombre	<i>Cowboy Duel – Disparar</i>
Descripción	El usuario es capaz de disparar.
Actores	Usuario
Precondiciones	El usuario está dentro del minijuego <i>Cowboy Duel</i> .
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario toca la pantalla una vez ha aparecido la señal de disparo. 2. En caso de haber disparado antes que el rival, gana la ronda y suma un punto. En caso contrario, pierde la ronda y no suma ningún punto.

Nombre	<i>Cowboy Duel – Fallar Disparo</i>
Descripción	El usuario es capaz de fallar el disparo si dispara antes de tiempo.
Actores	Usuario
Precondiciones	El usuario está dentro del minijuego <i>Cowboy Duel</i> .
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario toca la pantalla antes de la señal de disparo. 2. El personaje falla el disparo y pierde la única bala por lo que pierde la oportunidad de ganar la ronda.

6. Diseño de la solución

6.1 Introducción

La RAE define el término juego como un ejercicio recreativo o de competición sometido a reglas, y en el cual se gana o se pierde. Se puede considerar que los videojuegos como tal siguen esta definición al pie de la letra si lo aplicamos a un contexto electrónico.

Existe debate a la hora de definir las mecánicas de juego, sin embargo, tomaremos como referencia la definición de Miguel Sicart, en el artículo *Defining Game Mechanics* [6]: “*Game mechanics are methods invoked by agents for interacting with the game world*”. Esto quiere decir que las mecánicas son las habilidades que poseen los propios jugadores para interactuar con el entorno del juego.

Se puede tomar como referencia el juego *Super Mario Bros* (Figura 69) para observar una de las mecánicas más conocidas y utilizadas en los videojuegos: el salto. En esta saga de videojuegos un simple salto puede ser usado con diferentes propósitos: esquivar obstáculos, romper bloques o incluso derrotar enemigos. Por el contrario, existen juegos mucho más sencillos como por ejemplo *Flappy Bird*, cuya única mecánica es alzar el vuelo un poco al presionar la pantalla.



Figura 69. Salto en el videojuego *Super Mario Bros*

En *Blitz Party* se ha buscado conseguir unas mecánicas simples pero que resulten satisfactorias para el jugador. Seguidamente, se van a detallar los minijuegos junto a algunas de sus mecánicas desde el punto de vista del diseño.

6.2 Binky Pursuit

Binky Pursuit es un minijuego en el que el jugador toma el control de un pequeño zorro y deberá atrapar a los *Binkies* que aparecen cada poco tiempo. Sin embargo, no está solo. Deberá hacer frente a un rival con el que comparte objetivo y, por ello, deberá capturar más *Binkies* que su contrincante si desea alzarse con la victoria. Debido a la naturaleza de este minijuego ha sido necesario la implementación de mecánicas esenciales, como la del movimiento, entre otras.

6.2.1 Diseño de la inteligencia artificial

Para tener un rival en los minijuegos *Binky Pursuit* y *Cowboy Duel* en el modo de un jugador, ha sido necesario el diseño de una IA. Debido a la simpleza de los minijuegos esta IA no ha necesitado ser especialmente compleja. La más interesante es la utilizada por el rival, el zorro azul, en el minijuego *Binky Pursuit* ya que utiliza un algoritmo *pathfinding* para poder identificar e ir a capturar a los *Binkies*.

Este algoritmo es capaz de calcular el recorrido más corto entre dos puntos, algo muy similar a lo que realiza el algoritmo de Dijkstra en grafos con pesos. En este caso se ha buscado la ruta más corta entre la posición actual del zorro y la posición actual del *Binky* más cercano. Al estar los *Binkies* en movimiento, es importante estar constantemente recalculando este valor para crear una sensación de persecución. Al recalcularse este valor, si aparece otro *Binky* más cercano, la IA cambiará de objetivo y perseguirá a este nuevo *Binky*. En la Figura 70 se muestra la ruta que va a tomar la IA para acercarse al *Binky* mediante un *gizmo* de *Unity* y en la Figura 71 se puede apreciar la máquina de estados de esta IA.



Figura 70. Captura mostrando la ruta calculada por el algoritmo *pathfinding*



Figura 71. Máquina de estados de la IA del rival en *Binky Pursuit*

6.2.2 Movimiento del Jugador

En un minijuego como este en el que los jugadores deben perseguir y atrapar ciertos elementos que se mueven por un terreno, la mecánica de movimiento es esencial. Al ser un juego 2D nuestros personajes únicamente podrán moverse por los ejes X e Y. Como ya se ha comentado anteriormente, la comodidad de los controles táctiles de los dispositivos móviles es inferior a la que ofrecen periféricos de entrada como, por ejemplo, mandos con *joysticks*. Sin embargo, se decidió hacer uso de un joystick virtual como control del personaje debido al *feedback* de los usuarios. Este movimiento está basado en el que realizan los personajes en juegos 2D del género *beat'em up*, donde los personajes únicamente se pueden mover por el eje X pero tienen una sensación de profundidad. Además, los *sprites* de los personajes se voltean al cambiar de dirección. Ejemplos de estos juegos son *Final Fight* o *Scott Pilgrim vs. The World* (Figura 72).



Figura 72. Captura del videojuego *Scott Pilgrim vs. The World*

6.2.3 Captura de Binkies

Esta mecánica hace referencia a la habilidad del jugador de poder capturar a los *Binkies* para así sumar puntos. Para la primera versión, se ideó que la captura de los *Binkies* ocurriera de manera automática cuando los personajes entraban en contacto con estos enemigos. El problema consistía en la falta de *feedback* al usuario y, por ello, se decidió añadir un botón para ejecutar esta mecánica

para que el jugador sintiera que él tenía el control de la captura. De este modo, cuando el personaje del jugador está en contacto con algún enemigo y se presiona el botón de captura, el enemigo debería desaparecer y se debería actualizar la puntuación.

6.3 Whack-A-Mole

En este minijuego los jugadores han de golpear a los topos que aparecen de los distintos agujeros antes de que se vuelvan a enterrar. En contraste al minijuego anterior, el jugador no toma el control de un personaje como tal, sino de una herramienta, el mazo, y esta es la responsable de impactar con los enemigos. En el modo de un jugador, se debe superar una puntuación establecida, mientras que en el modo multijugador se compete por ver quién golpea a los topos primero y, por ende, quién obtiene mayor puntuación. Para crear este minijuego se han necesitado ciertas mecánicas relacionadas con los controles táctiles, la detección al golpear a estos enemigos y la generación de estos enemigos.

6.3.1 Movimiento y golpeo

Este minijuego presenta unos controles más sencillos que *Binky Pursuit* debido a la jugabilidad más simple que se presenta. Al no controlar un personaje como tal, y simplemente tener que usar las funciones táctiles para poder golpear a los enemigos, no ha sido necesario diseñar ningún elemento de control adicional.

La mecánica del movimiento hace referencia a la habilidad del jugador de crear un mazo al tocar la pantalla. La mecánica del golpeo es la que permite que el mazo interactúe con los enemigos que aparecen y otorgue la puntuación. Estas mecánicas por separado tienen una utilidad limitada, sin embargo, juntas dotan al juego de una jugabilidad completa.

A la hora de diseñar estas mecánicas, se tuvieron en cuenta ciertos factores para obtener la mejor jugabilidad posible. El factor más importante era la velocidad de pulsado de la pantalla. Se pensó que se debía limitar la velocidad a la que se podía pulsar la pantalla para que no se pudiera simplemente tocar rápidamente la pantalla para ganar. Sin embargo, el tiempo entre pulsaciones debía ser relativamente corto para que el minijuego no resultara aburrido.

Unas mecánicas similares a estas son las que presenta el videojuego *Fruit Ninja* (Figura 73), un juego de móvil en el que debemos deslizar el dedo por la pantalla táctil para cortar la fruta. Al tocar la pantalla aparece una estela que representa un cuchillo y al tocar la fruta, esta se corta.





Figura 73. Captura del videojuego *Fruit Ninja*

6.4 Cowboy Duel

En este minijuego, volvemos a ponernos en la piel del pequeño zorro y nos enfrentamos a un rival: el zorro azul. La jugabilidad simula un duelo de vaqueros del antiguo oeste donde ganaba el más rápido en desenfundar su pistola. En este caso los jugadores deben ser más rápidos que el rival tocando la pantalla para disparar cuando aparezca la señal. Dada la simpleza de este minijuego se describirá la mecánica de disparo.

6.4.1 Disparo

Este minijuego presenta una jugabilidad similar a un *quick time event* (Figura 74), una escena de un videojuego donde el jugador debe reaccionar rápidamente a la aparición de algún símbolo en pantalla. En este caso, cuando aparece la señal con el mensaje “shoot” el jugador debe simplemente tocar la pantalla para disparar y obtendrá la victoria si lo hace antes que su rival.

Durante esta fase se tuvo en cuenta que los jugadores podrían simplemente tocar la pantalla sin parar hasta que apareciera la señal para ganar. Para evitar este tipo de artimañas, le añadimos a esta mecánica la habilidad de fallar el disparo si se toca la pantalla antes de la señal. Si esto ocurre, el jugador pierde la bala y ya no puede disparar una vez aparezca la señal. De este modo, se consigue un sentimiento de tensión y adrenalina al tener que esperar y saber que no se puede fallar.



Figura 74. *Quick time event* del videojuego *Uncharted 3*

7. Unity3D

En este capítulo se entra en mayor detalle en la funcionalidad del motor de juego *Unity*. Se introducen ciertas características para facilitar el entendimiento de cómo funciona esta herramienta y comprender cómo se ha utilizado para el desarrollo de las mecánicas de juego.

7.1 Interfaz de Usuario

El editor de *Unity* nos presenta una interfaz muy flexible basada en ventanas. El usuario puede realizar diversas acciones con estas ventanas, como moverlas de posición o de tamaño para personalizarse el editor a su gusto y trabajar de la manera más cómoda posible. Además, al crear un proyecto nuevo, el editor abre directamente unas pestañas básicas con las que se puede empezar a desarrollar el videojuego, pero el usuario puede en cualquier momento añadir o quitar ventanas a su gusto. En la Figura 75 se puede observar el editor y a continuación se comentarán las ventanas principales.



Figura 75. Editor de *Unity*

Proyecto

Esta ventana muestra el contenido del proyecto actual y actúa como un explorador de archivos con funcionalidades comunes como eliminar o duplicar elementos. *Unity* denomina a todos estos elementos que se pueden incorporar al juego *assets*, y esta es la ventana principal desde la que los incorporaremos al juego arrastrándolos a otras ventanas. Para añadir nuevos *assets* podemos agregarlos desde el explorador de archivos del sistema operativo a la carpeta del proyecto.

Escena

Esta ventana nos enseña la propia escena que se ha creado para el videojuego. Aquí se pueden observar todos los elementos que forman parte de esta escena, y se pueden manipular ciertas características como su posición o rotación. Todos estos cambios se aplican de una manera inmediata, facilitando así la toma de decisiones a la hora de crear y diseñar las diferentes escenas que forman parte del videojuego.

Juego

Desde esta ventana nos ponemos en la piel del jugador para ver el resultado de ejecutar la escena que hemos creado. Desde esta vista podemos realizar diversas operaciones como, por ejemplo, cambiar la resolución para comprobar si el juego se adapta correctamente o ver diferentes estadísticas referentes al audio y los gráficos para observar el rendimiento. Los componentes *Camera* de *Unity* son los responsables de ofrecer el punto de vista deseado.

Jerarquía

En esta ventana podemos observar todos los elementos que forman parte de la escena actual. Al igual que desde la ventana del proyecto, podemos realizar diversas operaciones con los elementos, en este caso denominados *gameobjects*. Destaca de esta ventana la habilidad de poder crear parentesco entre diversos elementos, agrupándolos, heredando propiedades del elemento padre. Esto resulta especialmente útil a la hora de tener que realizar un cambio sobre varios *gameobjects* al mismo tiempo. Se pueden insertar todos dentro de un padre y realizar el cambio sobre este objeto en lugar de realizarlo sobre cada *gameobject* individualmente.

Inspector

Cuando se quiere ver o editar las propiedades de algún *gameobject*, *asset* o *prefab* se hará uso de esta ventana. Desde el inspector, además de poder modificar ciertos parámetros como los valores de la posición en los tres ejes, podemos añadir nuevos componentes a los objetos y, asignarles valor a las variables de visibilidad pública o serializadas de los scripts sin necesidad de tener que abrir el código desde un IDE.

Animator

Desde esta ventana podemos crear máquinas de estado que reflejen el comportamiento de los objetos y asignarle a cada estado una animación. Los cambios de estado pueden ocurrir debido a la activación de ciertas variables que actúan de disparador o de ciertas condiciones que el usuario puede elegir.

Audio Mixer

Esta ventana nos permite realizar múltiples cambios a las fuentes de audio del videojuego. Se puede utilizar para mezclar varias fuentes de audio o corregir el *pitch*. Sin embargo, su uso más común es el de cambiar la atenuación del volumen para poder crear unos ajustes de sonido funcionales.

7.2 Escenas

Las escenas de Unity nos permiten crear una división de una manera lógica del videojuego. Se pueden considerar como contenedores de una sección del juego completo. Para mejorar la comprensión se pone el ejemplo de un juego de tres niveles. Unity es muy flexible a la hora de trabajar y no existen unos estándares para el uso de las escenas, con lo cual se podría crear este juego en una única escena. Sin embargo, normalmente se crea una división en la que cada nivel es una escena diferente y al completar un nivel se pasa a la siguiente. Además, las escenas también se pueden utilizar para contener menús u otros tipos de contenidos. En el caso de *Blitz Party* se ha creado una escena por cada minijuego en modo de un solo jugador, una por cada minijuego en modo multijugador, una para el menú principal, una para la ventana de registro e inicio de sesión y finalmente una escena utilizada para inicializar componentes. Podemos observar estas escenas en la Figura 76.

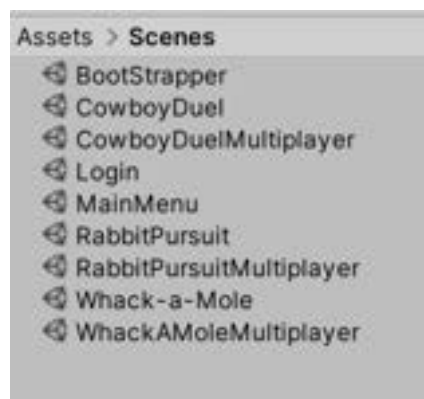


Figura 76. Escenas de *Blitz Party*

Al crear una nueva escena, esta se crea siempre con dos componentes básicos: una *Main Camera*, necesaria para proporcionar un punto de vista al renderizar la escena, y una *Directional Light*, la cual aporta la iluminación a la escena que acabamos de crear. A partir de aquí se puede construir la escena con total libertad.

7.3 GameObjects y Prefabs

Se denominan *gameobjects* todos los objetos que componen una escena de *Unity*. Sin embargo, estos objetos no son más que contenedores de componentes que les aportan una variedad de propiedades y comportamientos. Sin estos componentes, los *gameobjects* no hacen nada por sí solos; carecen de funcionalidad y de propiedades visuales. Se pueden crear este tipo de objetos a través de código o haciendo uso del propio editor, desde donde Unity proporciona unos *gameobjects* básicos listos para usar como podemos ver en la Figura 77.

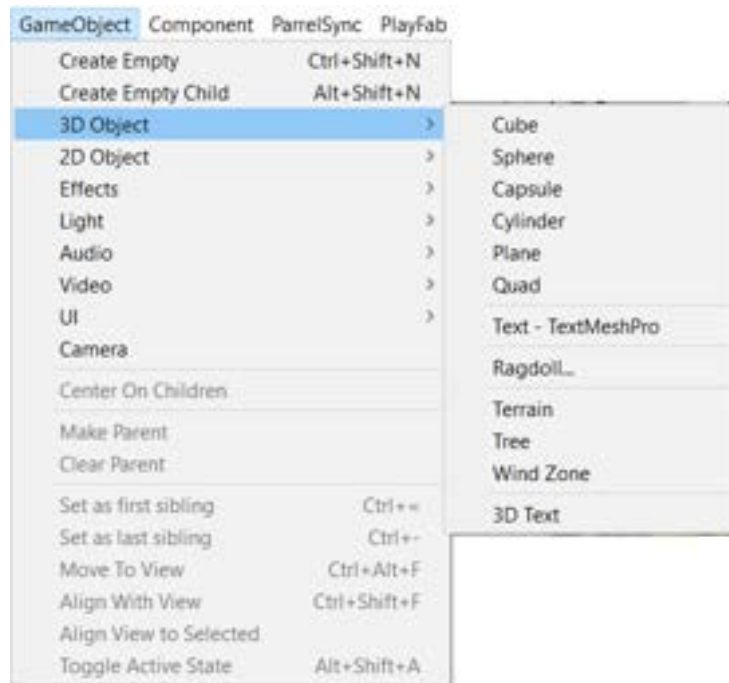


Figura 77. Listado de *GameObjects* ofrecidos por *Unity*

En el caso de querer reutilizar *gameobjects* en *Unity* se puede duplicar el objeto como tal para tener varias copias, no obstante, si se tiene la intención de editar estos objetos, se deberá hacer uno por uno al ser copias independientes. Para solventar este problema se utilizan los *prefabs*. Estos son un tipo de *asset* que permite guardar un *gameobject* con todas sus propiedades y componentes. Este *prefab* hace de plantilla de este objeto y a partir de él se pueden instanciar todos los objetos de este tipo que sean necesarios. Además, si realizamos cualquier cambio en el *prefab* original, este se verá reflejado en todas sus instancias. Para este proyecto se han usado estos *prefabs* para contener todos los objetos que deben ser reutilizados. Como podemos ver en la Figura 78, estos constan principalmente de los enemigos, ya que aparecen múltiples de ellos a la vez en pantalla.

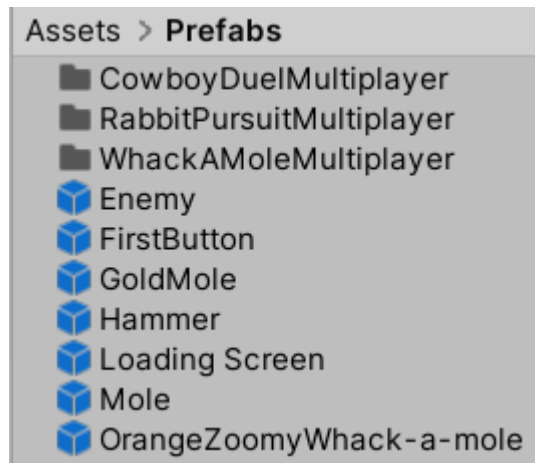


Figura 78. Prefabs utilizados en *Blitz Party* (iconos en azul)

7.4 Componentes

Como se ha mencionado en el epígrafe anterior, los *gameobjects* requieren componentes que les otorguen ciertas propiedades y funcionalidad. Unity ofrece un gran abanico de componentes de usos muy variados. A continuación, se explicarán los más destacables para la implementación de las mecánicas.

Transform

Este es el componente básico, ya que todos los *gameobjects* lo necesitan. Es el responsable de proporcionar la posición, la rotación y la escala de los objetos en la escena. Podemos modificar los valores de estas propiedades desde la ventana del inspector (Figura 79) o desde código.

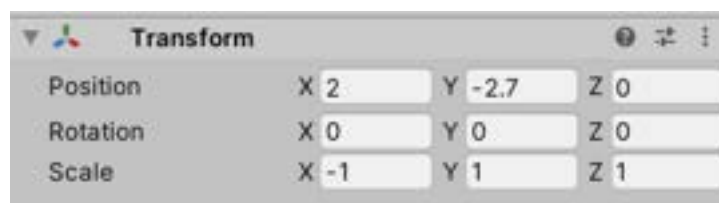


Figura 79. Vista del componente *Transform* desde el inspector de *Unity*

Rigidbody

El componente de *Unity Rigidbody* (Figura 80) permite colocar un objeto bajo la influencia del motor de físicas y de esta manera poder aplicarle fuerzas. Se pueden modificar ciertos parámetros como la masa o la gravedad para personalizar los efectos de las físicas al objeto que contiene el componente. Al ser un juego 2D se han hecho uso de los componentes *Rigidbody2D*.



Figura 80. Vista del componente *Rigidbody 2D* desde el inspector de *Unity*

Collider

Estos componentes dotan a los objetos de una forma para poder controlar las colisiones físicas. Objetos con este componente (Figura 81) podrán colisionar con otros objetos que también los tengan o simplemente podemos configurarlos para detectar estas colisiones sin necesidad de tener que chocar. Como se verá en el capítulo sobre la implementación, los *colliders* han sido especialmente útiles para la creación de las mecánicas de los minijuegos *Binky Pursuit* y *Whack-A-Mole*.

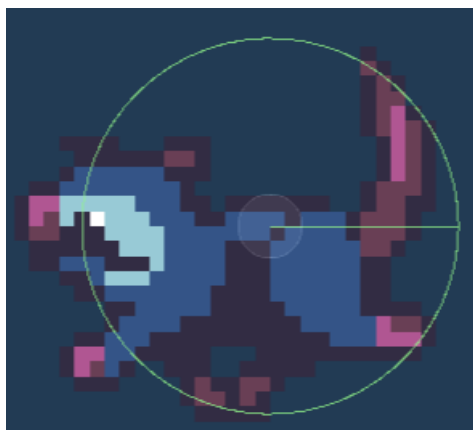


Figura 81. Sprite de un enemigo con el *collider* visible

Sprite Renderer

Este componente permite darle a un *gameobject* la apariencia de una imagen para ser usados en escenas. En este caso, al ser un juego completamente en 2D, los *gameobjects* que hacen referencia a personajes y enemigos hacen uso de estos componentes para que tengan la apariencia de sus correspondientes *sprites*.

Animator

Este componente hace uso de un *animator controller* para poder darles animación a los objetos. Para editar este comportamiento se hace uso de la ventana *Animator* que se ha mencionado anteriormente.

7.5 Scripts

Unity hace uso de scripts para dotar a los objetos de nueva funcionalidad. Además de añadir los propios componentes proporcionados por *Unity*, los scripts nos permiten incorporar las características y comportamientos propios de nuestro juego a estos objetos. De este modo, el desarrollador únicamente está limitado por su propia creatividad.

Es importante mencionar ciertas funciones que han sido necesarias para la implementación de los minijuegos:

- ***Awake***: esta función se llama únicamente cuando se está inicializando el objeto que contiene el script sin tener en cuenta si el propio script esta activo. Es especialmente útil para inicializar variables o estados antes de que comience la aplicación.
- ***Start***: esta función se ejecuta una única vez al igual que el método *awake* pero solo si el script esta activo.
- ***Update***: este método se llama una vez por fotograma y es especialmente útil para realizar un chequeo constante y así poder actualizar ciertas propiedades cuando se cumpla alguna condición.
- ***FixedUpdate***: función similar al *Update*, sin embargo, se ejecuta dependiendo de la frecuencia del sistema de físicas en lugar de por cada fotograma y por tanto es útil para realizar todo lo relacionado con las físicas de los objetos y así mantener la sincronización con el motor.
- ***OnEnable/OnDisable***: estas funciones se llaman cuando se activa o desactiva el objeto que contiene el script.



- ***OnDestroy***: función que se ejecuta durante el último fotograma de existencia de un objeto. Útil para casos en los que queremos que ocurra algo cuando destruyamos un objeto, por ejemplo, sumar un punto al derrotar a un enemigo.

También han sido esenciales ciertos métodos relacionados con los componentes *collider* y *animator* como son:

- ***OnTriggerEnter***: esta función se ejecuta cuando se tienen dos *colliders* y uno de ellos está marcado como *trigger*, y detecta que los *colliders* se acaban de solapar.
- ***OnTriggerExit***: similar a la función anterior, pero en este caso se ejecuta cuando se detecta que los *colliders* ya no están solapados.
- ***OnStateMachineExit***: esta función se llama cuando se alcanza el estado *exit* de la máquina de estados del *animator controller* de un objeto.

8. Implementación

A continuación, se van a comentar distintas técnicas de programación utilizadas para la implementación de las mecánicas que forman el juego. Además, se hace hincapié en el desarrollo de estas mecánicas desde un punto de vista técnico.

8.1 Patrón Object Pool

A la hora de generar los enemigos, *Unity* nos ofrece la función *Instantiate* para crearlos. Sin embargo, si hay muchos enemigos, ejecutar esta función cada vez puede resultar en una alta carga para la CPU. Por este motivo, se decidió hacer uso del patrón de diseño *Object Pool*. Este patrón consiste en tener un grupo de objetos preparados para su uso desde el comienzo en lugar de tener que crear y destruir los propios objetos bajo demanda. De esta manera se consigue optimizar el uso de recursos. Podemos ver un ejemplo de su uso en la Figura 82.

De todos modos, es cierto que en nuestros minijuegos la cantidad de enemigos que aparecen no es tan alta como para presentar una carga muy elevada en las CPU y en dispositivos actuales no existe gran diferencia de rendimiento cuando se utiliza el patrón. Sin embargo, al abrir el juego para un público general el único requerimiento es una versión mínima de *Android* 4.4, por lo tanto, consideramos imprescindible optimizar el juego para que funcione lo mejor posible en todo tipo de dispositivos sin importar la potencia.

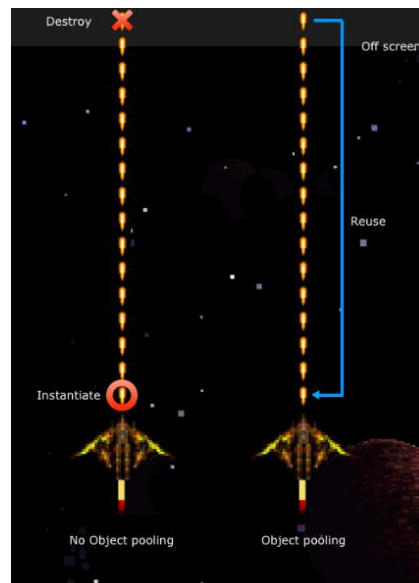


Figura 82. Ejemplo visual del patrón *object pool* obtenida de raywenderlich.com

8.2 Modularidad

Uno de los objetivos de este proyecto era el de crear una base sólida utilizando elementos modulares y reutilizables para facilitar la implementación de futuros minijuegos. Esto se considera de vital importancia ya que los juegos basados en minijuegos requieren de una gran variedad de mecánicas y sistemas ya que cada uno de ellos debe ser considerado como una experiencia completa. Por este motivo, es imprescindible que los componentes que se creen al comienzo del desarrollo sean reutilizables para así reducir el esfuerzo necesario para la implementación de los siguientes minijuegos.

En el desarrollo de *Blitz Party* se han seguido estos principios y por ello la gran mayoría de elementos y sistemas son completamente reutilizables. Esto es gracias a que se ha dividido el software en bloques independientes y se ha hecho que las clases y funciones realicen una única función en lugar de juntar múltiples funcionalidades dentro de una misma clase. Esto ha dado lugar a tener una gran cantidad de *scripts* de pocas líneas de código lo cual mejora la modularidad.

Ejemplos de elementos que se han reutilizado en el estado actual del juego son el sistema del cronometro que se puede ver tanto en los minijuegos *Binky Pursuit* como *Whack-A-Mole*, el sistema de puntuación que aparece en los tres minijuegos o el sistema de sonido que se utiliza tanto en los menús como en los minijuegos. Para crear muchos de estos sistemas se ha hecho uso de interfaces para reducir al máximo el acoplamiento entre clases. Podemos observar un ejemplo de estas interfaces en la Figura 83.

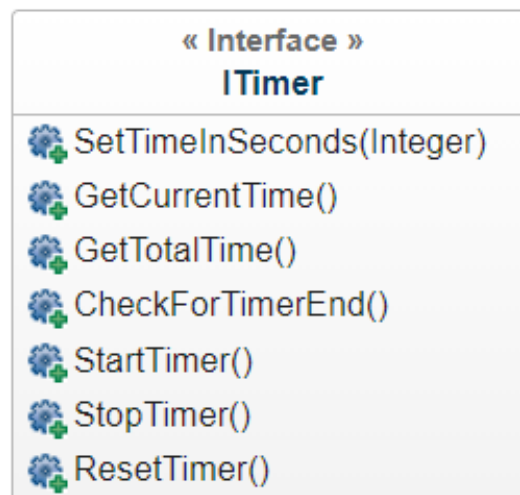


Figura 83. Interfaz del cronómetro

8.3 Binky Pursuit

8.3.1 Movimiento del Jugador

Para el movimiento del jugador se decidió utilizar el componente de *Unity Rigidbody2D*, para poder aplicarle fuerzas. Una vez obtenida la dirección del joystick que tiene seleccionada el jugador, se utiliza este componente para aplicarle la fuerza al personaje en esa misma dirección, simulando de esta manera el movimiento.

Para obtener la dirección del *joystick* usamos el método *ControlJoystickInput()*, que se llama cada fotograma desde el método *Update* de *Unity*, para obtener las magnitudes en ambos ejes y de este modo poder aplicar fuerzas en las direcciones seleccionadas. También actualizamos la variable de tipo *float* usada para actualizar el estado del componente *Animator* y que el personaje realice la animación de correr cuando se detecta movimiento en el *joystick* y vuelva a su estado inactivo cuando este cese. Podemos observar esta pequeña máquina de estados en la Figura 84.

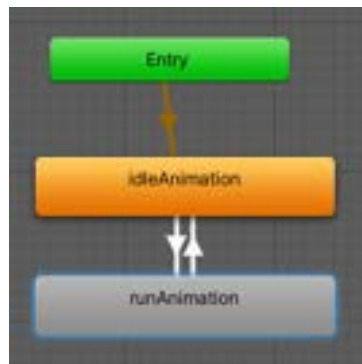


Figura 84. Máquina de estados del personaje en *Binky Pursuit*

Finalmente, el método *Move()* (Figura 85) es el encargado de efectuar la fuerza sobre el *Rigidbody* del personaje. Se crea y se normaliza un vector con los parámetros de la dirección obtenidas del *joystick* usado para obtener el vector con el destino al que se desea mover el personaje. Además, se ha usado el propio método de *Unity SmoothDamp()* para suavizar el movimiento al permitir pasarle el tiempo aproximado en el que se alcanzará el vector destino. Se ha tenido también en cuenta el cambio de dirección en el eje X comprobando los valores de dirección originales y este se refleja en la orientación del *sprite* del personaje gracias el método *Flip()*.

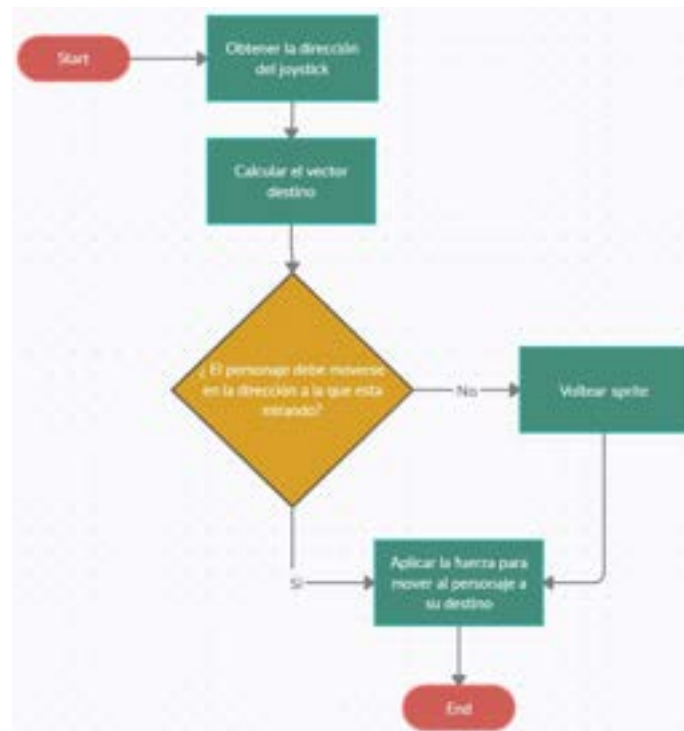


Figura 85. Diagrama de flujo de la función *Move()*

8.3.2 Generador de Binkies

La idea principal de este minijuego es que aparezca un número aleatorio de enemigos de cualquiera de las madrigueras para que los jugadores no puedan predecir donde saldrán. En este sistema de generación de enemigos es donde entra en juego el patrón *object pool* que se ha descrito en el capítulo anterior. Al entrar al minijuego, se crean todas las instancias de los enemigos en estado inactivo y cuando los enemigos aparecen se activan como se puede observar en la Figura 86. Estos enemigos se desactivan y vuelven a la *pool* cuando algún jugador los captura o cuando vuelven a esconderse.



Figura 86. Captura de *Binky Pursuit* con los enemigos visibles activos en la *Object Pool*

Para realizar esto se ha vuelto a hacer uso de la función *Update* de *Unity*, para tener una comprobación constante de ciertos valores y tener un control de la generación de estos enemigos. Para desarrollar esta mecánica, se usa una variable *time* la cual representa el tiempo que debe pasar antes de que se genere la siguiente oleada de enemigos. A su vez, existe la variable *spawnTime*, que representa el ratio de aparición de los enemigos.

Al comienzo del minijuego este valor se establece en 5 segundos y cada oleada aumenta para que se acelere la generación de enemigos con el paso tiempo y hacer que el minijuego sea más dinámico. No solo aumenta el ratio de aparición, sino que además, dependiendo del tiempo restante, se aumenta la probabilidad de que aparezcan más *Binkies* por oleada. Para esto, se realiza diversas comparaciones del tiempo total del minijuego con el tiempo actual. Una vez se obtiene el número de enemigos de la oleada se ejecuta la función *SpawnEnemy()* con este valor como parámetro. Esta función es la encargada de activar a los enemigos y hacer que aparezcan de una madriguera aleatoria. Podemos observar el diagrama de flujo de este algoritmo en la Figura 87.

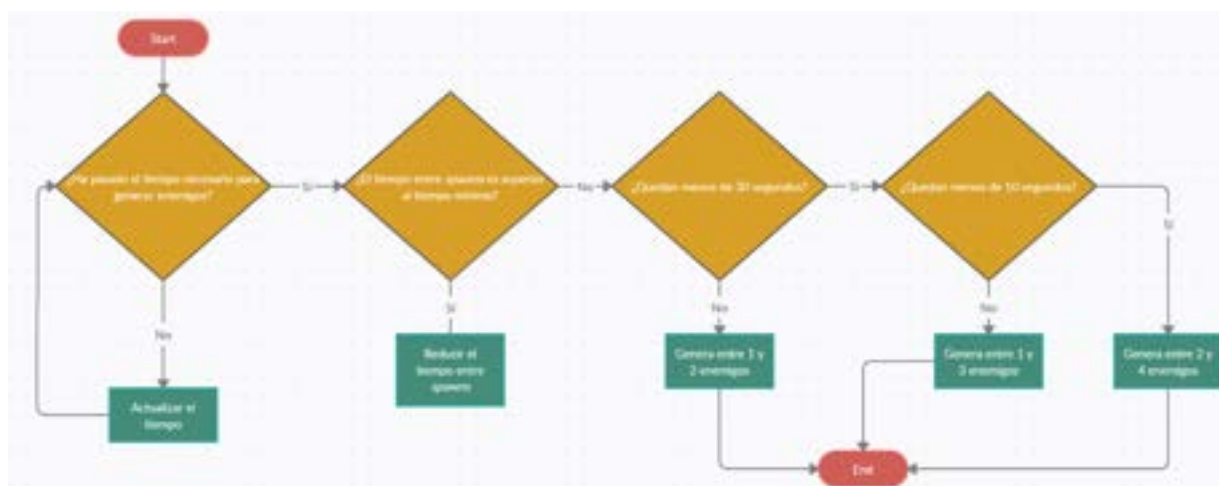


Figura 87. Diagrama de flujo del algoritmo responsable de la generación de *Binkies*

8.3.3 Captura de *Binkies*

Tanto los jugadores como los *Binkies*, contienen componentes de tipo *collider* junto a otros de tipo *RigidBody 2D* para que se reflejen las físicas como se ha explicado anteriormente. La captura de los *Binkies* ocurre cuando el *collider* del personaje se solapa con el *collider* de uno de los *Binkies* y se presiona el botón de capturar.

Para conseguir que esta mecánica funcione correctamente se han creado dos variables. La primera, *touchingBinky*, es de tipo *boolean* y es la encargada de controlar si el jugador está en contacto con un *Binky*. La segunda variable es la referencia al propio objeto con el que el jugador ha colisionado. Se han controlado estas variables con los métodos *OnTriggerEnter2D* y *OnTriggerExit2D*, que se ejecutan cuando los *colliders* entran en contacto y cuando dejan de estar en contacto, respectivamente.

Al presionar el botón de captura se comprueba si el valor de *touchingBinky* es true, es decir, si el *collider* del personaje está en contacto con el *collider* de uno de los *Binkies*, y, en caso positivo, se desactiva el objeto con el que ha colisionado y se suma un punto al jugador correspondiente.

8.4 Whack-A-Mole

8.4.1 Movimiento del mazo

Para lograr el efecto de golpeo se ha creado una función denominada *PerformAction()* (Figura 88) que es la responsable de poder atizar a los enemigos. Esta función se ejecuta cada fotograma y comprueba si existe algún toque en la pantalla y si el mazo no está en uso. En caso de que ambas condiciones se cumplan, se obtiene un vector con la posición exacta donde ha ocurrido el toque, se reproduce un efecto de sonido y se crea una instancia del objeto que representa el mazo. Este objeto, al ser creado, realiza una animación para mayor claridad visual del efecto del golpeo.

Una vez se ha creado el mazo, se ejecuta una corrutina, es decir, una función que nos permite ejecutar código de manera asíncrona, con el objetivo de poder tener el control de cuándo el mazo está en uso. Esta función es la responsable de liberar el mazo después de usarlo para prepararlo para el siguiente golpe. De esta manera, los jugadores no pueden simplemente tocar la pantalla muy rápido para ganar, sino que deben calcular cuando deben golpear.

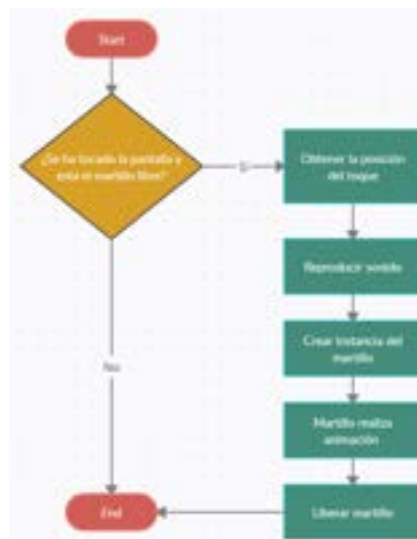


Figura 88. Diagrama de flujo de la función *PerformAction()*

8.4.2 Generador de enemigos

En este minijuego, la generación de enemigos se realiza de manera similar al anterior, sin embargo, existen dos características que las diferencian. La primera es el hecho de que existe más de un tipo de enemigo como podemos ver en la Figura 89 y la segunda es que cuando un enemigo se genera no puede generarse otro en el mismo lugar hasta que el primero se haya desactivado, ya que los enemigos simplemente aparecen de los agujeros y no se mueven por el mapa.



Figura 89. Enemigos del minijuego *Whack-A-Mole*

Debido a los distintos tipos de enemigos, se ha creado una *Object Pool* única para cada tipo. En este caso, cada tipo de enemigo tiene un ratio de *spawn* similar a el caso de los *Binkies* pero en este caso cada tipo tiene uno diferente. De esta manera, los enemigos que más aparecen son los topos que dan un único punto en contraste a los topos dorados que dan cinco. Al igual que en el minijuego anterior, estos ratios de *spawn* aumentan con el tiempo para acelerar el juego y darle dinamismo a la partida.

Para solventar el problema de que no aparezcan sobre el mismo agujero, cada tipo de generador de enemigos tiene una referencia a un único array que contiene si el agujero esta ocupado. Al intentar generar un enemigo, se recorre este array y se comprueba si hay un agujero libre. En caso positivo el enemigo aparece, en caso negativo el generador debe esperar a que uno de los agujeros se libere para así poder generar al enemigo. En este minijuego los agujeros se liberan cuando se golpea a un enemigo o cuando finaliza la animación de este sin ser golpeado.

8.4.3 Golpeo de enemigos

Como acabamos de ver en la sección anterior, existen tres tipos de enemigos en este minijuego. Uniendo las mecánicas que se terminan de explicar, obtenemos la habilidad de golpear con el martillo, y podemos ver cómo aparecen y desaparecen los distintos enemigos. Sin embargo, este golpeo debe reflejarse tanto en el estado de las criaturas como en el del minijuego.

De manera similar a cómo se ha realizado la mecánica de captura de *Binkies*, se han añadido componentes del tipo *collider* a los enemigos y al mazo para poder manejar las colisiones. Cuando el mazo colisiona con algún objeto se obtiene una referencia al componente *animator* del objeto con el que ha chocado. A partir de aquí, se compara el *Tag*, es decir, la etiqueta del objeto, para poder determinar el tipo de *gameobject* al que se ha golpeado. Con esto en cuenta, se reproduce

el efecto de sonido correspondiente al enemigo, se le otorgan al jugador los puntos correspondientes y se activa el *trigger*, un interruptor responsable de cambiar a la animación del enemigo golpeado.

En lugar de simplemente desactivar el objeto como se hace en *Binky Pursuit*, en este minijuego se hace uso de las máquinas de estados del componente *animator* para que, al golpear a los enemigos, estos realicen una pequeña animación antes de desaparecer. Para lograr esto, se ha creado una simple máquina de estados (Figura 90) en la que el enemigo, por ejemplo, el topo, sale de la madriguera en el estado *MoleAnimationUp* y cuando esta animación termina, se vuelve a esconder con el estado *MoleAnimationDown*. La transición entre estos dos estados depende únicamente del tiempo, sin embargo, cuando se activa el *trigger* que se acaba de mencionar, se interrumpe esta transición y se pasa al estado *MoleHit*. Aquí se ejecuta la animación de golpeo de los enemigos y cuando esta termina, al igual que cuando finaliza la animación de *MoleAnimationDown*, se pasa al estado final: *Exit*. Hemos usado este estado para que cuando se llegue a él por cualquiera de los dos caminos posibles, es decir, cuando han terminado sus animaciones, los enemigos dejen de estar activos.

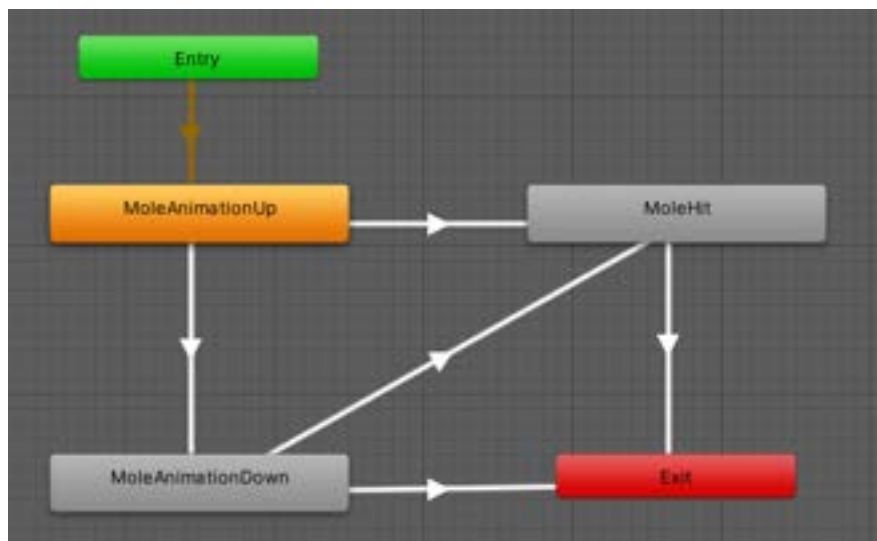


Figura 90. Máquina de estados del topo

8.5 Cowboy Duel

8.5.1 Disparo

Para implementar esta mecánica se han creado dos variables de tipo booleano para poder controlar ciertos momentos del minijuego. La primera variable *canShoot* coge el valor true cuando los jugadores ya pueden disparar. La segunda variable *hasShootAppeared* obtiene el valor true cuando ya ha aparecido la señal de disparo. Además, se ha añadido una función de auto disparo

para que en modo multijugador si ninguno de los dos jugadores dispara se fuerce a terminar la ronda.

Cuando la variable *canShoot* es true se comienza a esperar la respuesta del jugador. Esto ocurre cuando nos encontramos en el estado *WaitToShoot* de la máquina de estados que podemos observar en la Figura 91. En este punto comienza a salir la cuenta atrás y después de un tiempo aleatorio aparece la señal que pone a true la variable *hasShootAppeared*. Si el jugador presiona la pantalla antes de la señal, cuando esta variable este en *false*, fallará y perderá la única bala de la ronda. En caso contrario ganará únicamente si ha disparado antes que el rival. Una vez ha terminado la ronda, se reinician las animaciones y se vuelve a empezar hasta alcanzar los dos puntos.

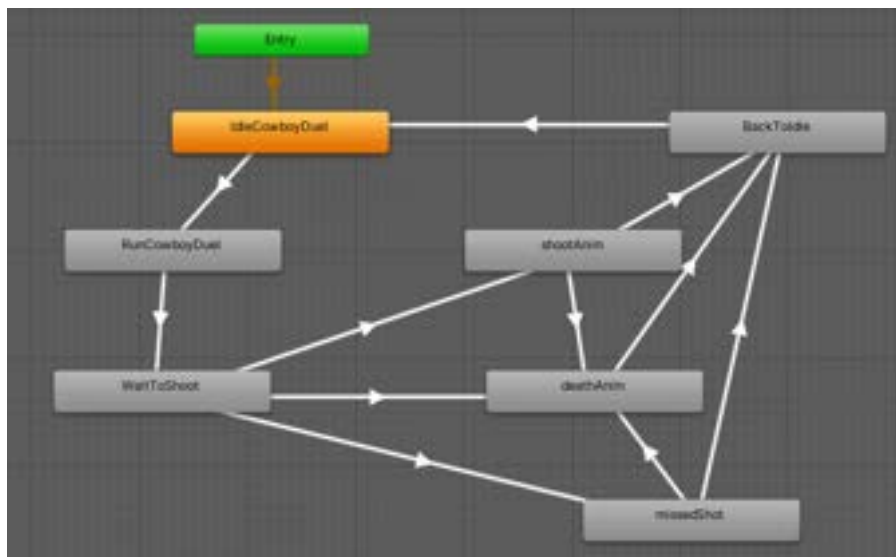


Figura 91. Máquina de estados personaje *Cowboy Duel*

9. Pruebas y resultados obtenidos

Una vez ha terminado el proceso de desarrollo se debe comprobar que las funcionalidades implementadas funcionen correctamente y se ajusten a las especificaciones originales. Para realizar estas operaciones se han hecho uso del sistema de pruebas de aceptación que proporciona *Worki* para poder probar si todo funcionaba correctamente haciendo uso de los dispositivos del equipo de desarrollo. Por otra parte, la opinión de los usuarios es esencial, sobre todo en lo referente a las mecánicas ya que éstas son las que marcan la diferencia entre algo divertido o tedioso para el jugador. Se han creado diferentes encuestas donde los jugadores pueden compartir sus opiniones respecto al juego para de este modo obtener *feedback* fiable.

9.1 Validación de requisitos

En la fase de *Especificar Requisitos* de cada *sprint*, se deben establecer las pruebas de aceptación asociadas al cambio que realizará la unidad de trabajo en el comportamiento del producto. Como se puede observar en la Figura 92, para cada unidad de trabajo relacionado con la jugabilidad de los minijuegos se han creado múltiples pruebas de aceptación para comprobar cada mecánica o sistema de juego.






Lista de UT		Pruebas de Aceptación				
+ Nueva PA Listado de PA						
	Nombre PA	Última modificación de PA				
		Fecha modificación	UT	Sprint	Colaborador	Resultado
	Q	Q	Q	Q	Q	Q
⚡	5775 - Aparición de resultados	29/4/2021 19:18	2737	Sprint 2		✓
⚡	5774 - Puntuación al golpear enemigos	29/4/2021 19:16	2737	Sprint 2		✓
⚡	5773 - Golpear enemigo	29/4/2021 19:14	2737	Sprint 2		✓
⚡	5771 - Aparición y desaparición de los enemigos	29/4/2021 19:12	2737	Sprint 2		✓
⚡	5770 - Martillo aparece correctamente	29/4/2021 19:10	2737	Sprint 2		✓

Figura 92. Pruebas de aceptación del minijuego *Whack-A-Mole*

Una vez establecidas las pruebas de aceptación se debe implementar la funcionalidad de la unidad de trabajo, en este caso, las mecánicas de juego, siguiendo las pautas marcadas por estas pruebas. Una vez estas unidades de trabajo han terminado la fase de desarrollo, entran a la fase denominada *Aplicar Pruebas de Aceptación*, donde se prueba la funcionalidad y se comprueban los resultados.

Si todas las pruebas de aceptación se pasan correctamente se considera que la unidad de trabajo ha sido terminada, en caso contrario, la unidad de trabajo se devuelve a la fase anterior para corregir los errores que han surgido y se vuelven a ejecutar las pruebas. Estas fases se pueden observar en la Figura 93.

Colaborador	Actividad	Notas de actividad	Creación	Finalización	Finalizador	Tiempo registrado
	Terminar		11/05/2021 21:55	11/05/2021 21:55		
	Aplicar Pruebas de Aceptación		11/05/2021 19:20	11/05/2021 21:55		50m
	Programar		01/05/2021 17:46	11/05/2021 19:20		10h 30m
	Programar		03/05/2021 18:12	11/05/2021 19:18		9h 22m
	Esperar Sprint		29/04/2021 20:11	01/05/2021 17:46		0m
	Diseñar y estimar		29/04/2021 20:07	29/04/2021 20:11		30m
	Especificar Requisitos		29/04/2021 19:55	29/04/2021 20:07		1h
	Registrar		29/04/2021 19:44	29/04/2021 19:55		0m

Figura 93. Fases de una unidad de trabajo

9.2 Encuestas a usuarios

Como se ha explicado en el capítulo tres, tras la salida de los *MVPs*, se realizaron encuestas para obtener *feedback* de los usuarios que habían probado el juego. En estas encuestas se pregunta por diversos apartados del juego para obtener datos referentes al juego completo, sin embargo, se van a destacar las relacionadas con las mecánicas y la jugabilidad de *Blitz Party*.

En la Figura 43 se pregunta acerca de la curva de aprendizaje que presenta el juego y la mayoría de las respuestas contestaron con una nota de diez, es decir, consideraban que es muy fácil aprender a jugar al juego. Sin embargo, existen respuestas con notas inferiores que demuestran que ciertos jugadores se encontraron con problemas a la hora de jugar. En lo referente a la jugabilidad, para resolver este problema, se implementaron ligeros cambios a los controles como por ejemplo reajustar la sensibilidad del *joystick* en el minijuego *Binky Pursuit*.

En la Figura 42 y Figura 44 se preguntaba directamente sobre la dificultad del juego y si algún minijuego les parecía demasiado difícil. Estas preguntas fueron clave para el avance del proyecto ya que, en este punto, el juego constaba únicamente del modo de un jugador y jugar contra una IA demasiado difícil resulta frustrante para el jugador. Por el contrario, una IA demasiado fácil hace que resulte aburrido jugar y, por ello, se buscaba encontrar el equilibrio. Las respuestas se

ajustaron a lo que se buscaba con la mayoría de las respuestas entre el cuatro y el seis. No obstante, *Binky Pursuit* era sin duda el juego que presentaba la mayor dificultad para los jugadores debido a la IA. Por este motivo se decidió reajustar sus valores para rebajar esta dificultad.

Para el segundo MVP, con el multijugador de *Binky Pursuit* y *Whack-A-Mole* funcionando se repitieron estas últimas dos preguntas con los resultados que podemos ver en la Figura 94 y Figura 95.

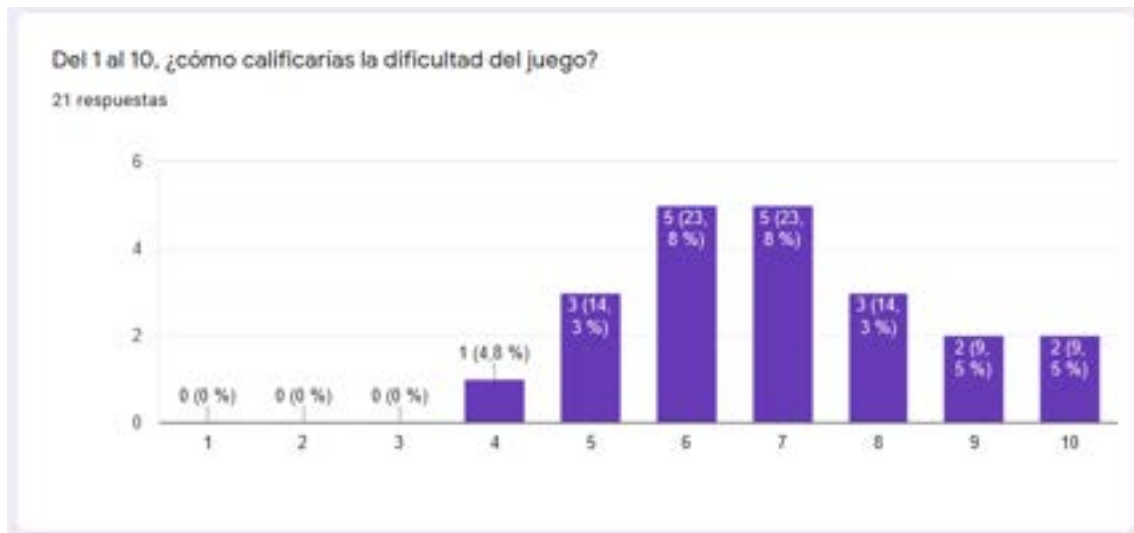


Figura 94. Pregunta dificultad general segundo MVP

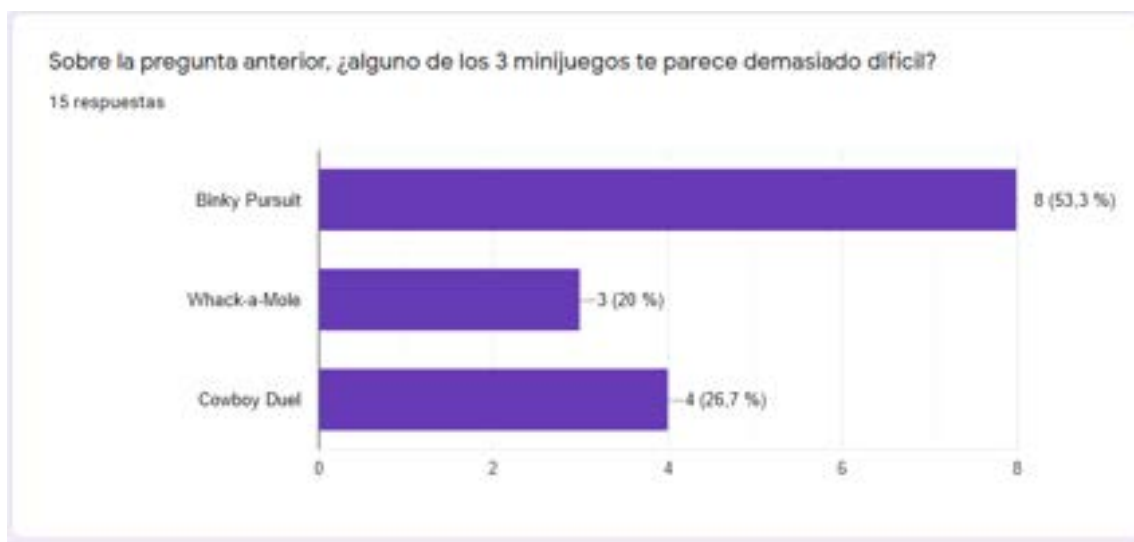


Figura 95. Pregunta dificultad de minijuegos segundo MVP

Podemos observar cómo a los resultados de la pregunta sobre la dificultad general del juego han tendido ligeramente más hacia el lado derecho, es decir, se considera que ha habido un ligero aumento de dificultad. Esto es posible, dado que ahora ciertos minijuegos se juegan contra otro rival humano y ya no basta ser mejor que la IA, sino que se debe jugar mejor que el contrincante.

Estos resultados parecen bastante aceptables y están dentro de los márgenes que se esperaba, con lo cual se considera que *Blitz Party* ha conseguido la dificultad deseada.

En la pregunta acerca de la dificultad excesiva de los minijuegos vemos cómo los resultados actuales presentan una forma similar a los resultados anteriores. A pesar de ello, cabe recalcar que ha habido una bajada de respuestas a esta pregunta y por ello ciertos encuestados no consideran que los minijuegos tengan una dificultad excesiva. Aun así, después de los cambios a la IA, *Binky Pursuit* sigue siendo el minijuego más complejo para los jugadores y se tienen pensados diversos cambios para rebajar esta dificultad.

Finalmente, se encuesta acerca de los propios controles de los minijuegos. Podemos ver los resultados en la Figura 96. Se observa cómo la gran mayoría de notas son altas, entre el ocho y el diez, y por tanto se considera que los controles son cómodos para el jugador y no es necesario realizar ningún cambio en este aspecto. Esto a su vez nos demuestra que la fuente de la dificultad de los minijuegos no son los controles.



Figura 96. Pregunta sobre la facilidad de control

10. Conclusiones

Este trabajo comienza con el objetivo principal de crear un videojuego multijugador para dispositivos móviles. Además, se buscó dar los primeros pasos en el mundo del emprendimiento con este proyecto, al pasar por todo el proceso que lo envuelve y finalizando con un producto competente listo para salir al mercado. Considero que, como equipo, hemos superado estos objetivos con éxito.

En lo referente a mis objetivos específicos considero que *Blitz Party* ha conseguido ser un videojuego divertido y sus minijuegos cumplen con la sencillez que se deseaba desde el inicio. Además, se ha cumplido el objetivo de crear un juego con componentes reutilizables para facilitar el futuro desarrollo de este.

El trabajo en equipo ha sido clave para el desarrollo de este proyecto. Hemos sido capaces de crear un ambiente de trabajo agradable a la vez que disciplinado que nos ha ayudado a sacar adelante el videojuego. Es importante recalcar que todo esto se ha conseguido con los miembros del equipo separados físicamente y mediante reuniones periódicas de manera telemática debido a la situación actual en la que vivimos.

Para el desarrollo de este proyecto han entrado en juego conocimientos que hemos adquirido a lo largo de toda la carrera y de áreas diferentes. Entre otros, han sido esenciales los conocimientos obtenidos en las asignaturas IPV (Introducción a la programación de videojuegos) y EDV (Entornos de desarrollo de videojuegos) ya que fueron mi primera toma de contacto con el motor *Unity* y donde descubrí el potencial de esta herramienta. Por supuesto, se deben mencionar las asignaturas PSW (Proceso de Software) y PIN (Proyecto de ingeniería del software) en las que aprendí sobre las metodologías ágiles y cómo aplicarlas al desarrollo de un proyecto.

Finalmente, se debe mencionar que a principios del desarrollo de este proyecto existió un quinto integrante del equipo que iba a ser el encargado del arte del videojuego ya que era nuestro punto débil al ser un equipo formado únicamente por desarrolladores. Sin embargo, por motivos ajenos al equipo actual, este nuevo integrante decidió abandonar el proyecto y, por tanto, se debe recalcar que exceptuando el diseño del *Zoomy*, el apartado artístico ha dependido de nosotros. Por suerte, hemos sacado algo positivo de todo esto y hemos desarrollado nuevas habilidades con herramientas de edición y diseño y nos hemos unido más como equipo al darnos cuenta de que sacar el proyecto adelante dependía únicamente de nosotros cuatro. Todo esto ha resultado una experiencia muy gratificante.

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).

RAE: Real Academia Española.

CPU: Central Processing Unit (Unidad de Procesamiento Central, en español).

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 del 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
- 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

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.....	105
Características generales	105
Características multijugador.....	105
Editor	105
Jugabilidad	105
4. El mapa del mundo.....	106
Visión general	106
Zonas temáticas	106
Parque en constante evolución	107
Sistema de renderizado.....	107
Cámara	107
Motor de videojuegos.....	107
5. Personajes del juego	108
Visión general	108
Creando un personaje	109
Personajes especiales para cada minijuego.....	109
6. Interfaz de usuario	109
Visión general	109
7. Herramientas	112
Visión general	112
Variedad de herramientas	112



8. Banda sonora y efectos de sonido.....	113
Visión general	113
Sonido 3D	113
9. Minijuegos	113
<i>Binky Pursuit</i>	113
<i>Whack-a-Mole</i>	116
<i>Cowboy Duel</i>	119
10. Juego para un jugador	122
Visión general	122
Historia.....	122
11. Juego multijugador.....	123
Visión general	123
Batalla de minijuegos	123
Máximo de jugadores	123
Servidores	123
Personalización	123
Internet	123
Persistencia	124
Guardado y cargado	124
12. Calendario de trabajo	124

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



Figura 97. 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 carnet

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 98 se observa el diseño inicial del menú principal del videojuego.



Figura 98. 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 99). 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.

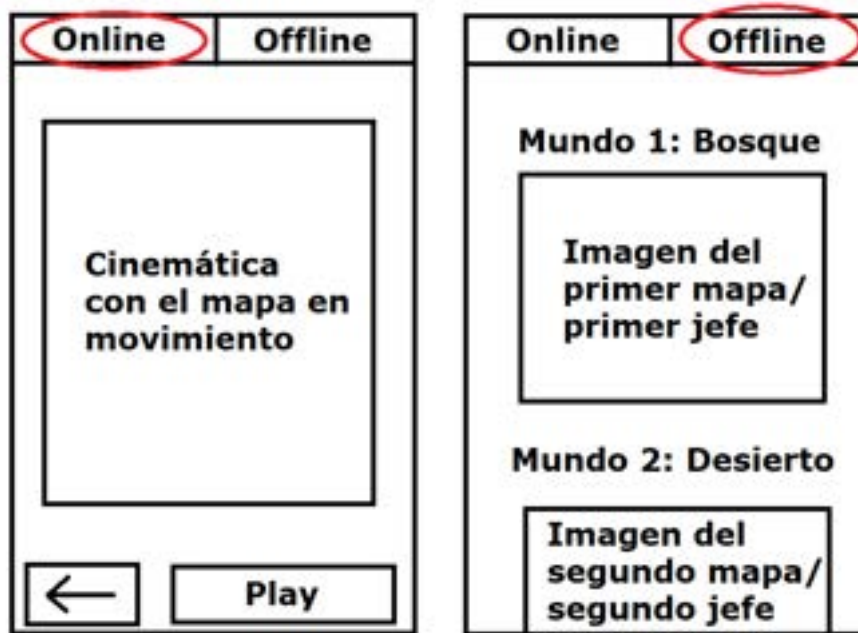


Figura 99. Pantallas de juego

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



Figura 100. 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 101) 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.

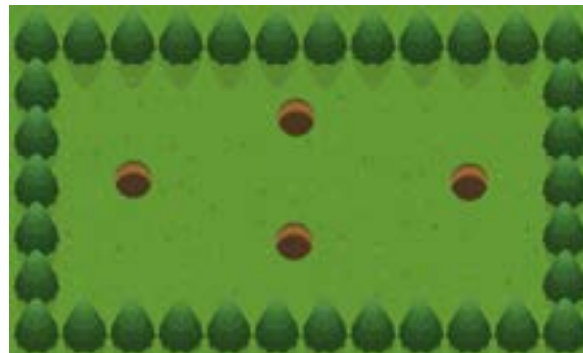


Figura 101. Escenario de *Binky Pursuit*

Interfaz de usuario

Como se aprecia en la Figura 102, 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.

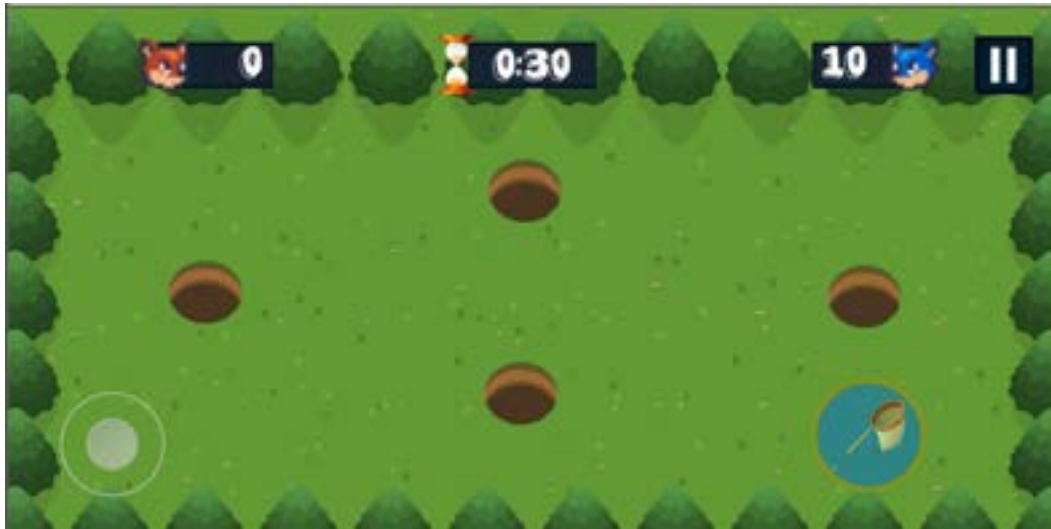


Figura 102. Interfaz de *Binky Pursuit*.

Personajes

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

- Zorros (Figura 103): Los personajes controlables por los jugadores.



Figura 103. Zorros de *Binky Pursuit*

- *Binkies* (Figura 104): 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.

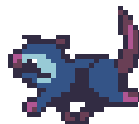


Figura 104. *Binkies* de *Binky Pursuit*

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 105) simula una llanura donde hay siete madrigueras de topos. En los bordes del mapa existen vallas que delimitan el mapa.

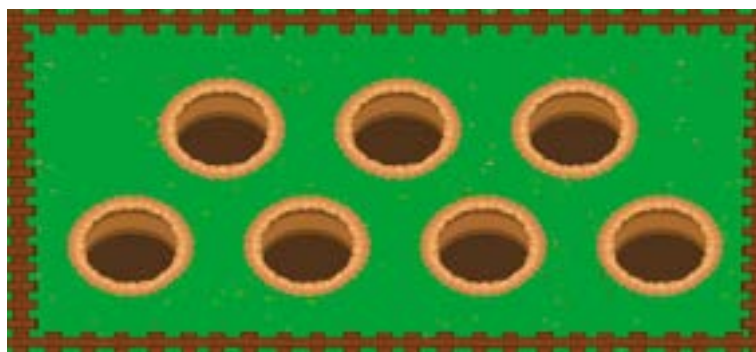


Figura 105. Escenario de *Whack-a-Mole*

Interfaz de usuario

Como se puede ver en la Figura 106, 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.

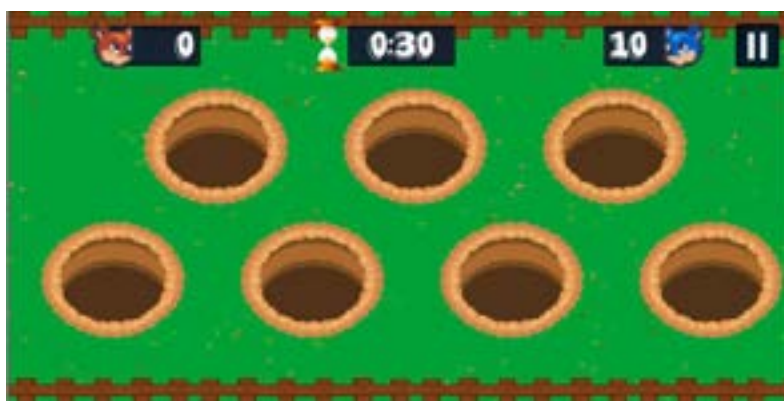


Figura 106. Interfaz de *Whack-a-Mole*

Personajes

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

- Martillo (Figura 107): es el personaje que controla el jugador. Simula el martillo real del arcade original *Whack-A-Mole* (Guacamole en España).



Figura 107. Martillo de *Whack-a-Mole*

- Topo (Figura 108): es el enemigo estándar. Golpearle con el martillo otorga un punto.



Figura 108. Topo normal de *Whack-a-Mole*

- Topo dorado (Figura 109): es un topo especial que tiene menor probabilidad de aparecer que el topo. Golpearle con el martillo otorga cinco puntos.



Figura 109. Topo dorado de *Whack-a-Mole*

- Zoomy (Figura 110): es el personaje del jugador. Golpearle con el martillo resta tres puntos.



Figura 110. 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 111) simula un desierto del lejano oeste donde los vaqueros solían tener este tipo de duelos.



Figura 111. Escenario de *Cowboy Duel*

Interfaz de usuario

Como se puede apreciar en la Figura 112, 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 112. Interfaz de *Cowboy Duel*

Personajes

En este minijuego participan los dos zorros, uno de cada jugador (Figura 113).



Figura 113. 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
Fase de Desarrollo				
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%
Fase de Testeo				
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%
Fase de Despliegue				
Listo para salida	Pablo	20/07/21	30/07/21	0%
Milestone: Listo para uso		20/07/21	30/07/21	0%

Figura 114. 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>
- [4] Video Game Sales Wiki: *Wario* <<https://vgsales.fandom.com/wiki/Wario>>
- [5] Adage: *How 'Dumb Ways to Die' won the Internet, became the no. 1 campaign of the year.* <<https://adage.com/article/special-report-the-awards-report/dumb-ways-die-dissected/245195>>
- [6] Miguel Sicart: *Defining Game Mechanics* <<http://gamestudies.org/0802/articles/sicart>>