



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Desarrollo de un videojuego en Unity:
implementación de jugabilidad multijugador y
servicio online

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Pablo Stemmer Fernández

Tutor: Francisco José Abad Cerdá

2020-2021

Resumen

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

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

En la parte de implementación de la funcionalidad multijugador, se realiza un enfoque al servicio *online* proporcionado al usuario, así como a una jugabilidad multijugador que permite la interacción de dos jugadores. Se desarrollan métodos y estrategias para llevar a cabo dicha cooperación entre usuarios. Ha sido necesario proporcionar una abstracción de código al mismo tiempo que aumentar la legibilidad y mantenibilidad del mismo a la hora de migrar al multijugador y para ello ha resultado beneficioso el uso de técnicas de *refactoring* y patrones de diseño.

Palabras clave: videojuego, *Unity*, minijuegos, emprendimiento, *multijugador*

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.

En la part de la implementació de la funcionalitat multijugador, es realitza un enfocament al servici online, proporcionat a l'usuari, així com una jugabilitat multijugador que permet la interacció de dos jugadors. Es desenvolupen mètodes i estratègies per a dur a terme aquesta cooperació entre usuaris. En la part de la implementació de la funcionalitat multijugador, es realitza un enfocament al servici online proporcionat a l'usuari, així com una jugabilitat multijugador que permet la interacció de dos jugadors. Ha sigut necessari proporcionar una abstracció del codi al mateix temps que augmentar la legibilitat i manteniment d'este a l'hora de migrar al multijugador i per a això ha resultat beneficiós l'ús de tècniques de refactoring i patrons de disseny.

Paraules clau: videojoc, Unity, minijocs, empenedoria, *multijugador*

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.

This section about the implementation of multiplayer functionality is focused on online services provided to the end-user, as well as multiplayer gameplay that allows the interaction of two players. Methods and strategies are developed to carry out the aforementioned cooperation among players. It was necessary to provide code abstraction and to increase readability and maintainability when it comes to migrating to multiplayer. To achieve that, the use of refactoring technics and design patterns has proven to be advantageous.

Keywords: video game, Unity, minigames, entrepreneurial, multiplayer.

Tabla de contenidos

1. Introducción	15
1.1 Motivación	16
1.2 Justificación del tema	17
1.3 Objetivos	17
1.4 Estructura de la memoria	18
2. Evaluación de la idea de negocio	19
2.1 Estado del arte	19
2.2 Estudio de mercado	22
2.2.1 Super Mario Party	24
2.2.2 Wario Ware	26
2.2.3 Among Us	28
2.2.4 Dumb Ways to Die	30
2.3 Análisis DAFO	32
2.4 Modelo de negocio y proyección económica	33
2.5 Lean Canvas	39
2.6 Conclusiones	42
3. Desarrollo de la idea de negocio	43
3.1 Mapa de características	44
3.2 Primer <i>Minimum Viable Product</i>	45
3.2.1 Desarrollo del <i>MVP</i>	46
3.2.2 Experimento	48
3.3 Segundo <i>Minimum Viable Product</i>	56
3.3.1 Desarrollo del <i>MVP</i>	57
3.3.2 Experimento	58
3.4 Despliegue del producto y <i>marketing</i>	61
4. Tecnologías utilizadas	63



4.1 Unity Engine	63
4.2 Visual Studio	63
4.3 Git/GitHub	64
4.4 Worki Process	65
4.5 Photoshop	65
4.6 Mirror	66
4.7 Azure PlayFab	66
5. Análisis y diseño del multijugador	68
5.1 Análisis	68
5.1.1 Brainstorming	68
5.1.2 Cuestionario	69
5.2 Diseño	70
5.2.1 Llamadas de Procedimiento Remotas	70
5.2.2 Comunicaciones entre scripts	73
Binky Pursuit	73
Whack-A-Mole	74
Cowboy Duel	75
6. Mirror	76
6.1 Introducción	76
6.2 <i>Mirror Framework</i>	76
6.2.1 <i>Network Manager</i>	76
6.2.2 <i>Network Discovery</i>	78
6.2.3 <i>Network Identity</i>	79
6.2.4 <i>Network Transform</i>	80
6.2.4 <i>Network Rigidbody</i>	81
6.2.4 <i>Network Animator</i>	82
7. Implementación del multijugador	83
7.1 Minijuegos	83
7.1.1 <i>Binky Pursuit</i>	83

NetworkManager	83
Elementos de la red	84
7.1.2 <i>Whack-A-Mole</i>	89
NetworkManager	89
Elementos de la red	90
7.1.3 <i>Cowboy Duel</i>	94
NetworkManager	94
Elementos de la red	94
8. Resultados	96
8.1 Encuestas	96
8.2 Pruebas de Aceptación	97
9. Conclusiones	99
10. Trabajo futuro	100
11. Apéndices	101
11.1 Listado de abreviaturas, siglas y acrónimos	101
11.2 Lista de recursos gráficos	101
11.3 <i>Game Design Document</i> (GDD)	103
1. Historial de diseño	106
Versión 1.00	106
Versión 2.00	106
Versión 3.00	106
2. Visión global del juego	106
Filosofía	106
Preguntas frecuentes	106
3. Conjunto de características	107
Características generales	107
Características multijugador	107
Editor	108
Jugabilidad	108



4. El mapa del mundo	108
Visión general	108
Zonas temáticas	108
Parque en constante evolución	109
Sistema de renderizado	109
Cámara	109
Motor de videojuegos	110
5. Personajes del juego	110
Visión general	110
Creando un personaje	111
Personajes especiales para cada minijuego	111
6. Interfaz de usuario	111
Visión general	111
7. Herramientas	114
Visión general	114
Variedad de herramientas	114
8. Banda sonora y efectos de sonido	115
Visión general	115
Sonido 3D	115
9. Minijuegos	115
<i>Binky Pursuit</i>	115
Descripción	115
Historia	115
Controles	115
Escenario	116
Interfaz de usuario	116
Personajes	117
Efectos visuales y sonoros	117
Dinámica del juego	117

Condición de victoria	117
<i>Whack-A-Mole</i>	118
Descripción	118
Historia	118
Controles	118
Escenario	118
Interfaz de usuario	118
Personajes	119
Efectos visuales y sonoros	120
Dinámica del juego	120
Condición de victoria	120
<i>Cowboy Duel</i>	121
Descripción	121
Historia	121
Controles	121
Escenario	121
Interfaz de usuario	121
Personajes	122
Efectos visuales y sonoros	122
Dinámica del juego	122
Condición de victoria	123
10. Juego para un jugador	123
Visión general	123
Historia	123
11. Juego multijugador	124
Visión general	124
Batalla de minijuegos	124
Máximo de jugadores	124
Servidores	124



Personalización	124
Internet	124
Persistencia	125
Guardado y cargado	125
12. Calendario de trabajo	125
12. Referencias bibliográficas	127

Listado de figuras

Figura 1. Logo de Blitz Party	15
Figura 2. Enlace QR de descarga de Blitz Party	16
Figura 3. Lanzamiento de jabalina en Olympic Decathlon	19
Figura 4. Partida de Bomberman Party Edition	20
Figura 5. Personaje personalizable de la Nintendo Wii	21
Figura 6. Captura del videojuego Buzz!: el Gran Reto	21
Figura 7. Cronología juegos party o de minijuegos	22
Figura 8. Tabla comparativa de productos competidores.....	23
Figura 9. Portada de Super Mario Party	24
Figura 10. Modo tablero de Super Mario Party.....	24
Figura 11. Minijuego de Super Mario Party.....	25
Figura 12. Portada de Wario Ware Gold.....	26
Figura 13. Microjuego de Wario Ware Gold	27
Figura 14. Cinemática del modo historia de Wario Ware Gold	27
Figura 15. Portada de Among Us	28
Figura 16. Impostor asesinando a un tripulante en Among Us	29
Figura 17. Jugador realizando una tarea en Among Us.....	30
Figura 18. Portada de Dumb Ways to Die.....	31
Figura 19. Fotograma de la canción Dumb Ways to Die	31
Figura 20. Microjuego de Dumb Ways to Die	32
Figura 21. Matriz DAFO de Blitz Party	33
Figura 22. Proyección económica año 1	35
Figura 23. Proyección económica año 2	36
Figura 24. Personal contratado.....	36
Figura 25. Proyección económica años 3 y 4.....	37
Figura 26. Proyección económica año 5	38
Figura 27. Resultado acumulado de ingresos.....	38
Figura 28. Lean Canvas de Blitz Party.....	41
Figura 29. Pitch Doc.....	44
Figura 30. Mapa de características de Blitz Party	45
Figura 31. Mapa de características del primer MVP.....	46
Figura 32. Minijuegos seleccionados del primer MVP.....	46
Figura 33. Captura del minijuego Binky Pursuit.....	47
Figura 34. Captura del minijuego Whack-A-Mole	48



Figura 35. Captura del minijuego Cowboy Duel	48
Figura 36. Tuit de promoción del primer MVP.....	49
Figura 37. Edad de los participantes de la encuesta	50
Figura 38. Tiempo que dedican a videojuegos los participantes de la encuesta	50
Figura 39. Porcentaje de participantes que juegan en dispositivos móviles.....	50
Figura 40. Horas que juegan los participantes de la encuesta en dispositivos móviles.....	51
Figura 41. Minijuego favorito de los encuestados	51
Figura 42. Puntuaciones de la curva de aprendizaje.....	52
Figura 43. Puntuaciones sobre la dificultad del juego.....	52
Figura 44. Opinión sobre dificultad excesiva de algún minijuego.....	53
Figura 45. Opiniones sobre la claridad de la interfaz gráfica.....	53
Figura 46. Opiniones sobre diseño y ambientación	54
Figura 47. Opinión sobre la música de los minijuegos	54
Figura 48. Puntuaciones generales de Blitz Party	55
Figura 49. Mapa de características del segundo MVP	56
Figura 50. Binky Pursuit multijugador.....	57
Figura 51. Whack-A-Mole multijugador.....	58
Figura 52. Pregunta rango de edad.....	59
Figura 53. Pregunta jugadores de dispositivos móviles	59
Figura 54. Pregunta horas jugadas en dispositivos móviles	60
Figura 55. Pregunta minijuego preferido	60
Figura 56. Pregunta puntuación general Blitz Party.....	61
Figura 57. Blitz Party en la Play Store	61
Figura 58. Países desde los que se ha descargado Blitz Party.....	62
Figura 59. Cuenta de Twitter de Swifter Games	62
Figura 60. Logo de Unity Engine.....	63
Figura 61. Logo del IDE Visual Studio.....	63
Figura 62. Logos de Git y GitHub respectivamente.....	64
Figura 63. Logo de Worki TUNE-UP Process.....	65
Figura 64. Logo de Adobe Photoshop.....	65
Figura 65. Logo de Mirror.....	66
Figura 66. Logo de Azure PlayFab	67
Figura 67. Brainstorming del multijugador	69
Figura 68. Encuesta análisis multijugador del minijuego Whack-A-Mole	70
Figura 69. Funcionamiento de un dispositivo Host.....	70
Figura 70. Funcionamiento comunicaciones remotas entre cliente y servidor en Mirror	72
Figura 71. Diagrama de dominio de los scripts de Binky Pursuit	73

Figura 72. Diagrama de dominio de los scripts de Whack-A-Mole.....	74
Figura 73. Diagrama de dominio de los scripts de Cowboy Duel.....	75
Figura 74. Componente Network Manager de Mirror.....	77
Figura 75. Componente Network Discovery de Mirror	79
Figura 76. Componente Network Identity de Mirror	79
Figura 77. Componente Network Transform de Mirror.....	80
Figura 78. Componente Network Rigidbody de Mirror.....	81
Figura 79. Componente Network Animator de Mirror	82
Figura 80. Componente RabbitPursuitNetworkManager.....	84
Figura 81. Componentes Network del objeto Player	85
Figura 82. Componentes de red de los objetos enemigos	86
Figura 83. Objeto ScoreController y sus componentes.....	87
Figura 84. Objeto RabbitPursuitManager y sus componentes	87
Figura 85. Objeto GUIController y sus componentes.....	88
Figura 86. Objeto EnemySpawner y sus componentes.....	89
Figura 87. Componente WhackAMoleNetworkManager	90
Figura 88. Prefab del jugador con sus componentes.....	91
Figura 89. Prefab del mazo con sus componentes.....	92
Figura 90. Prefab de tipo de enemigo topo estándar	93
Figura 91. Prefab del spawner de enemigos de tipo topo comunes.....	93
Figura 92. Componente CowboyDuelNetworkManager.....	94
Figura 93. Prefab jugador CowboyDuel y sus componentes	95
Figura 94. Satisfacción sistema de batalla 1vs1 del multijugador.....	96
Figura 95. Pruebas de aceptación del multijugador de Binky Pursuit.....	97
Figura 96. Pruebas de aceptación de Whack-A-Mole.....	98
Figura 97. Prueba de aceptación de Cowboy Duel	98
Figura 98. Personajes de Blitz Party	111
Figura 99. Mockup menú principal Blitz Party.....	113
Figura 100. Pantallas de juego	114
Figura 101. Herramientas de los minijuegos.....	114
Figura 102. Escenario de Binky Pursuit.....	116
Figura 103. Interfaz de Binky Pursuit	116
Figura 104. Zorros de Binky Pursuit.....	117
Figura 105. Binkies de Binky Pursuit	117
Figura 106. Escenario de Whack-A-Mole.....	118
Figura 107. Interfaz de Whack-A-Mole.....	119
Figura 108. Martillo de Whack-A-Mole	119



Figura 109. Topo normal de Whack-A-Mole.....	119
Figura 110. Topo dorado de Whack-A-Mole.....	119
Figura 111. Zoomy de Whack-a-Mole.....	120
Figura 112. Escenario de Cowboy Duel.....	121
Figura 113. Interfaz de Cowboy Duel.....	122
Figura 114. Personajes de Cowboy Duel.....	122
Figura 115. 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 de juego del videojuego.
- Marc Garcia Ferrer: Desarrollo del apartado *back-end* del videojuego.
- Pablo Stemmer Fernández: Desarrollo de las funcionalidades multijugador del videojuego, desarrollado en esta memoria.

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

1.1 Motivación

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

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

El motivo de la formación del equipo de desarrollo es que debido a la conexión previa entre sus integrantes y a la compenetración existente, se facilita la aplicación de metodologías ágiles.

Además, la conexión personal de cada uno con los videojuegos es objeto de motivación para la creación de un producto de calidad y del que sentirse orgulloso.

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

1.2 Justificación del tema

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

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

1.3 Objetivos

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

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

- Realizar dos *minimum viable product* o MVP con sus experimentos correspondientes, con la finalidad de obtener un producto completamente funcional y conseguir *feedback* de los jugadores.
- Indagar en el mundo del emprendimiento, llevando a cabo un proyecto de emprendimiento de principio a fin y utilizando metodologías ágiles para organizarlo y estructurarlo.
- Publicar el videojuego en la tienda digital *Play Store* de *Android* para facilitar la distribución y poder acceder a un mercado mayor.
- Diseñar y desarrollar una funcionalidad y servicio multijugador que permita la interacción de dos personas y un juego simultáneo en diferentes dispositivos.
- Profundizar en el diseño, funcionamiento e implementación de sistemas multijugador en videojuegos.



1.4 Estructura de la memoria

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

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

A continuación, se exponen las tecnologías utilizadas para el desarrollo del producto, así como el porqué de su elección y su impacto en el mismo. En las secciones siguientes, se indaga en la parte de análisis y diseño sobre cómo se va a llevar a cabo, seguido de la explicación del funcionamiento de *Mirror* y la propia implementación y desarrollo del multijugador para los minijuegos examinando como se han construido haciendo uso del *framework* de *Mirror* y terminando con una parte de pruebas realizadas a cabo para su posterior lanzamiento en el MVP 2.

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 videoconsolas más potentes como la *Nintendo 64* y la *Playstation*. Debido a esto, en el año 1998 se estrenaron dos juegos que definieron el futuro del género *party*, se trata de *Mario Party* y *Bomberman Party Edition*. *Mario Party* sentó las bases de lo que consideramos hoy en día un juego *party*. Este juego consiste en un tablero donde los jugadores tendrán que conseguir el mayor número de estrellas posibles. Para conseguir estas estrellas, deberán vencer a los otros

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

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

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



Figura 4. Partida de *Bomberman Party Edition*

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



Figura 5. Personaje personalizable de la *Nintendo Wii*

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



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

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

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

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

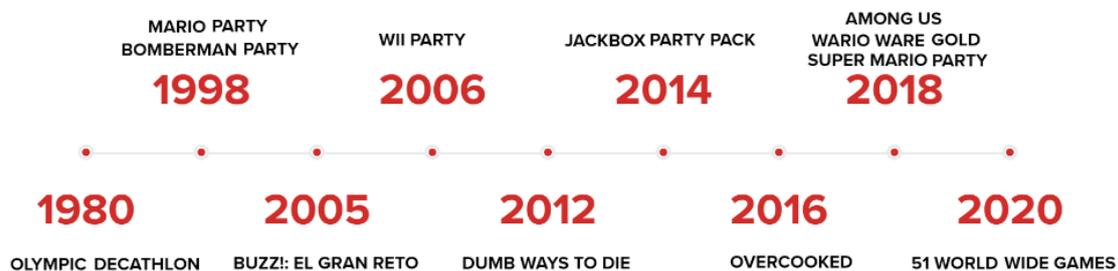


Figura 7. Cronología juegos *party* o de minijuegos

2.2 Estudio de mercado

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

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

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

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

Figura 8. Tabla comparativa de productos competidores

2.2.1 Super Mario Party

La saga *Mario Party*, cuya última entrega se observa 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 que los otros jugadores y estas monedas 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.

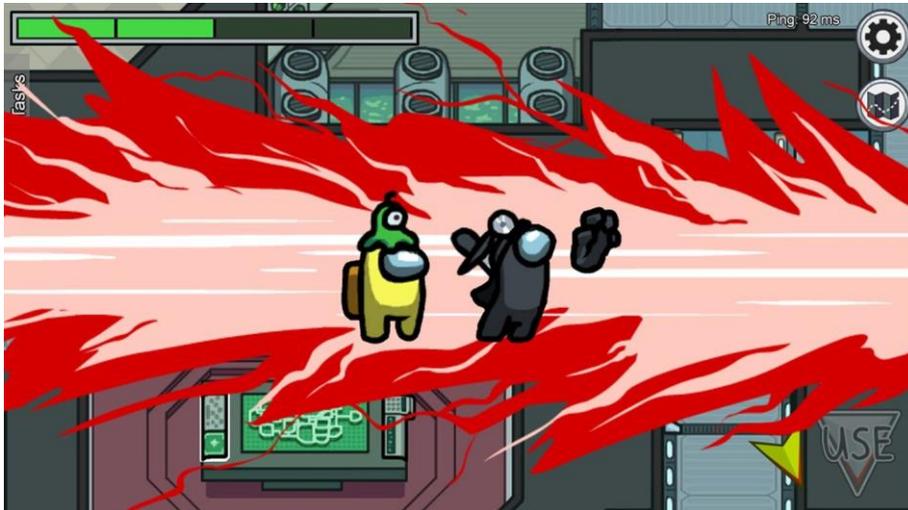


Figura 16. Impostor asesinando a un tripulante en Among Us

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].

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

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

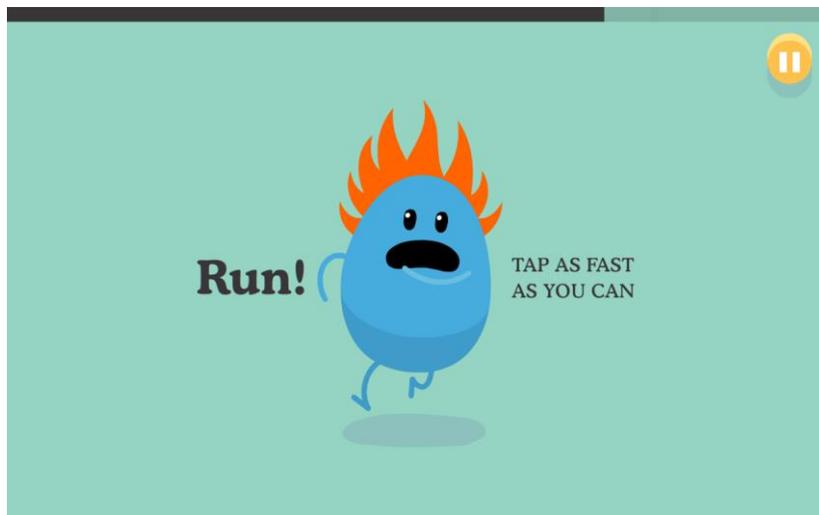


Figura 20. Microjuego de Dumb Ways to Die

2.3 Análisis DAFO

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

<p style="text-align: center;">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 style="text-align: center;">Amenazas</p> <ul style="list-style-type: none"> • Mercado con mucha competencia. • Funcionalidad limitada del servicio online gratuito.
<p style="text-align: center;">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 style="text-align: center;">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 tipo de coste monetario, siguiendo la concepción de que un juego gratuito atrae a más gente que otro de pago. No obstante, el juego cuenta con un sistema de micropagos con el que se adquiere una cantidad de monedas de juego, bellotas en este caso, a cambio de un precio asequible.



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

Por otra parte, los productos comprados en la tienda del juego son meramente estéticos, por lo que pagar más no dará a ningún jugador una ventaja real sobre su oponente. Como segunda forma de financiación, se prevé la introducción de anuncios en el juego, que se mostrarán en momentos específicos, como al inicio de la aplicación o al final de los minijuegos, y podrán ser eliminados si se paga la mejora de 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			
Número de nuevas descargas	Año 1 / T1	Año 1 / T2	Año 1 / T3
Número de nuevas descargas	20000	4000	4000
Dispositivos en los que estaba instalado el juego previamente	0	15000	13000
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 jugabilidad multijugador y servicio online

	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	3	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. Proyección económica año 5

La proyección económica se ha realizado a 5 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.).

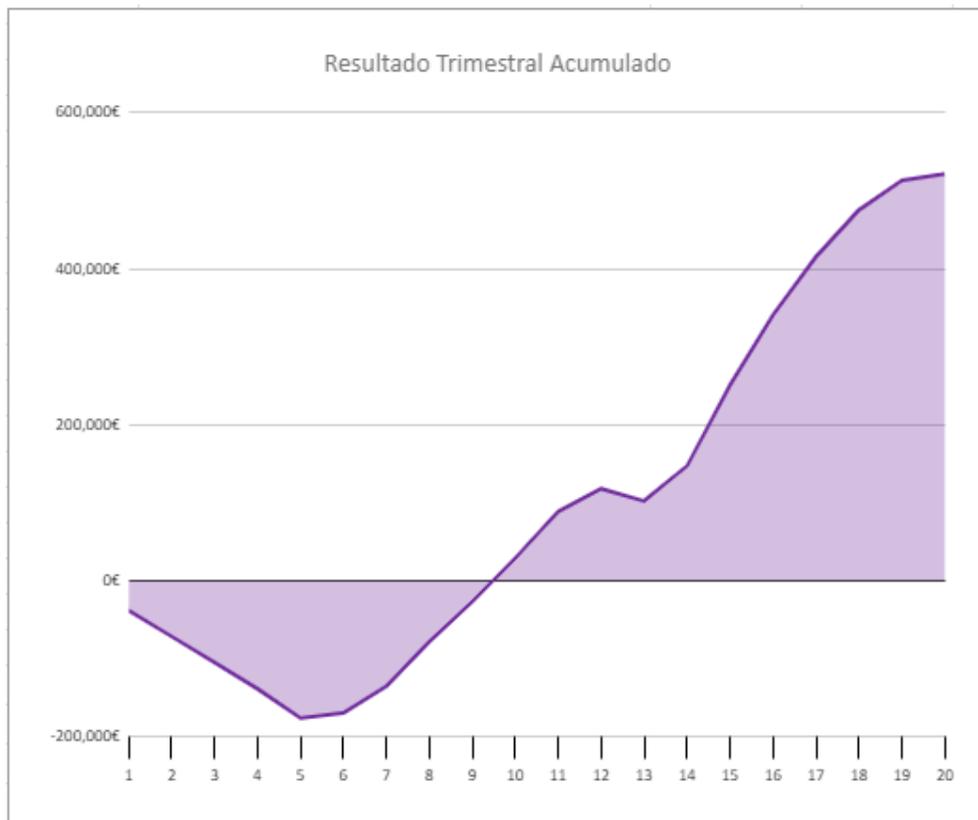


Figura 27. Resultado acumulado de ingresos

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 En el mercado de juegos móviles, tanto los juegos estilo party como los juegos estilo minijuegos son un terreno bastante inexplorado.</p> <p>Algunos posibles competidores son <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 party. Pero ninguno de ellos cumple todas estas características.</p>	<p>4 Solución 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 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 -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> <p>5 Canales - 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>	<p>1 Clientes -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.</p>
---	--	--	--	--

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 son dinámicos y muy divertidos, sobre todo cuando se pueden disfrutar con amigos. Se propuso la realización de un videojuego de estas características, que tuviese multijugador online y se pudiese instalar en dispositivos móviles.

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

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

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

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



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

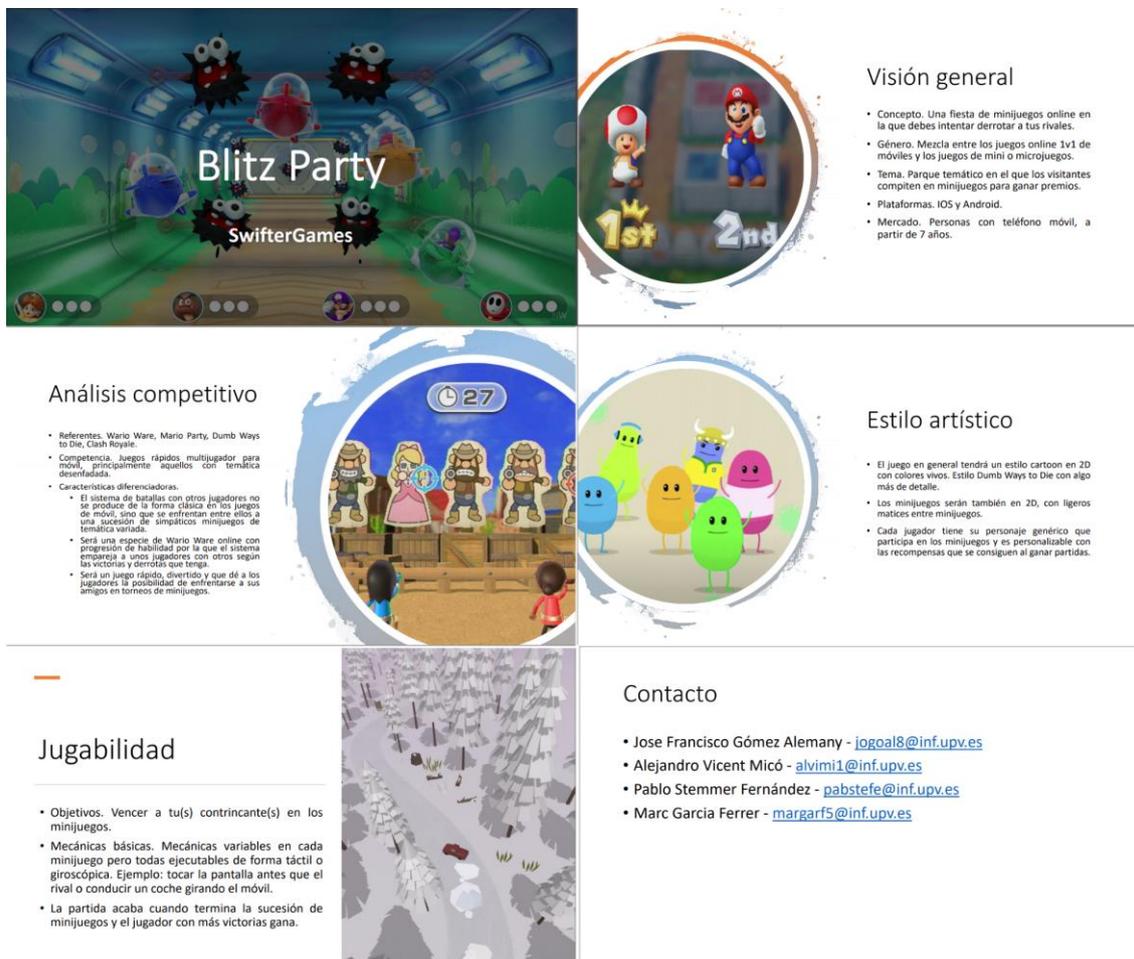


Figura 29. Pitch Doc

3.1 Mapa de características

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



Figura 30. Mapa de características de *Blitz Party*

3.2 Primer *Minimum Viable Product*

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

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

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



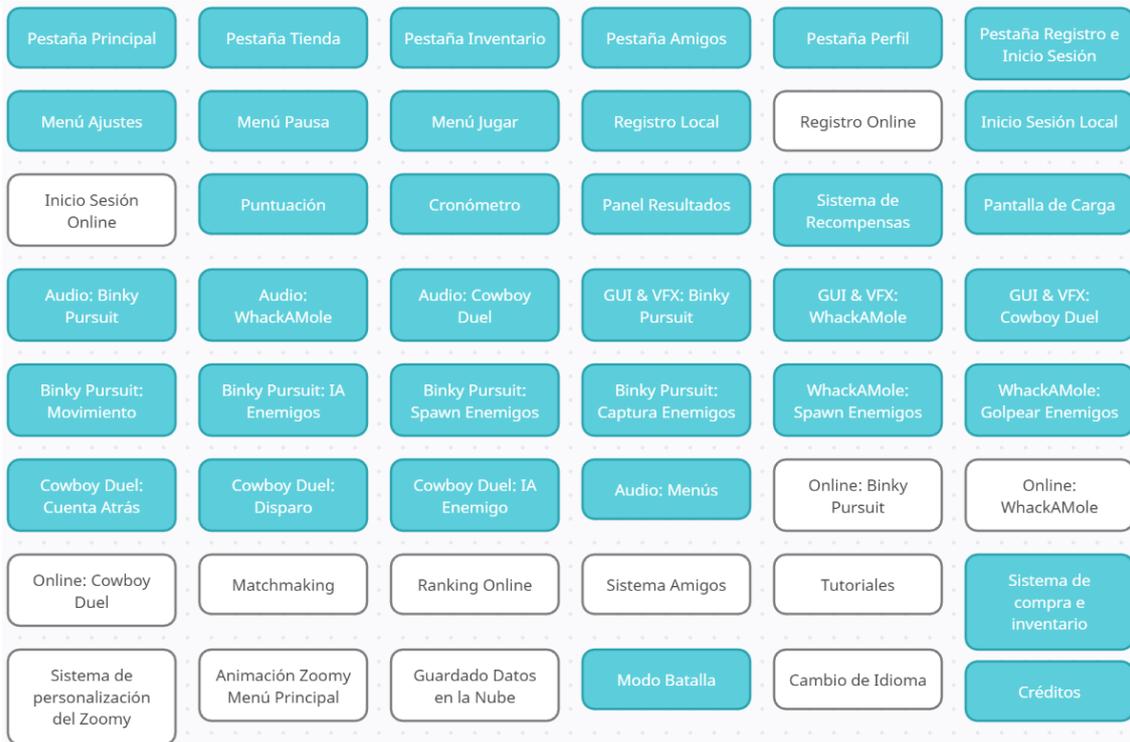


Figura 31. Mapa de características del primer MVP

3.2.1 Desarrollo del MVP

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



Figura 32. Minijuegos seleccionados del primer MVP

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

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



Figura 33. Captura del minijuego Binky Pursuit

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



Figura 34. Captura del minijuego Whack-A-Mole

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



Figura 35. Captura del minijuego Cowboy Duel

3.2.2 Experimento

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



Figura 36. Tuit de promoción del primer MVP

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

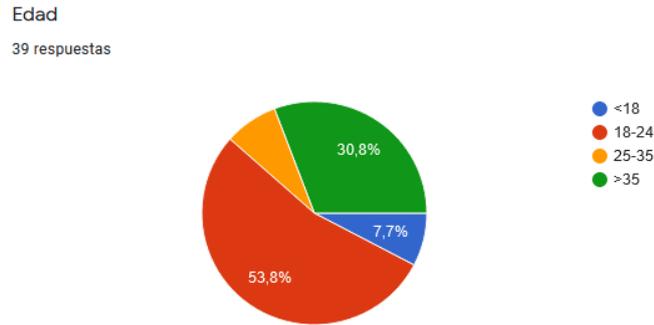


Figura 37. Edad de los participantes de la encuesta

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

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

39 respuestas

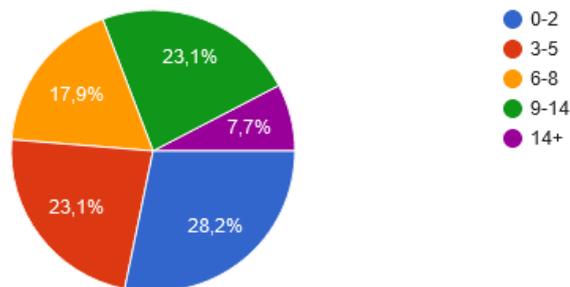


Figura 38. Tiempo que dedican a videojuegos los participantes de la encuesta

¿Sueles jugar a videojuegos en dispositivos móviles?

39 respuestas

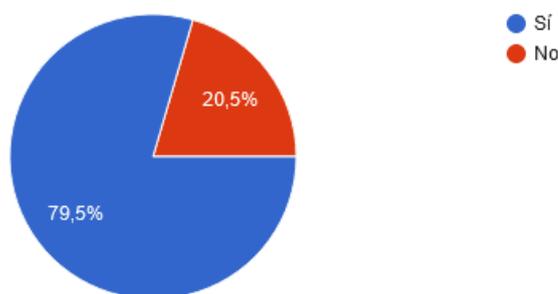


Figura 39. Porcentaje de participantes que juegan en dispositivos móviles

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

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

33 respuestas

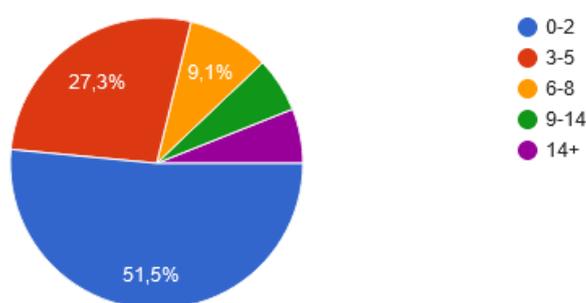


Figura 40. Horas que juegan los participantes de la encuesta en dispositivos móviles

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

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

39 respuestas

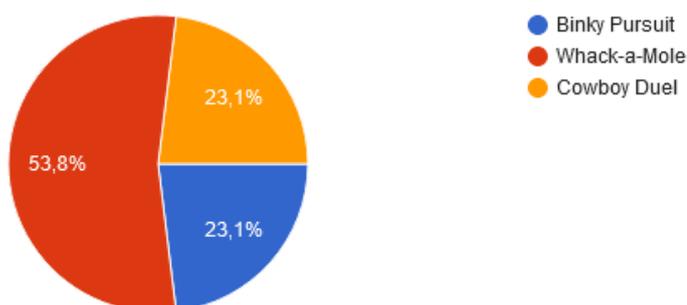


Figura 41. Minijuego favorito de los encuestados

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

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

39 respuestas

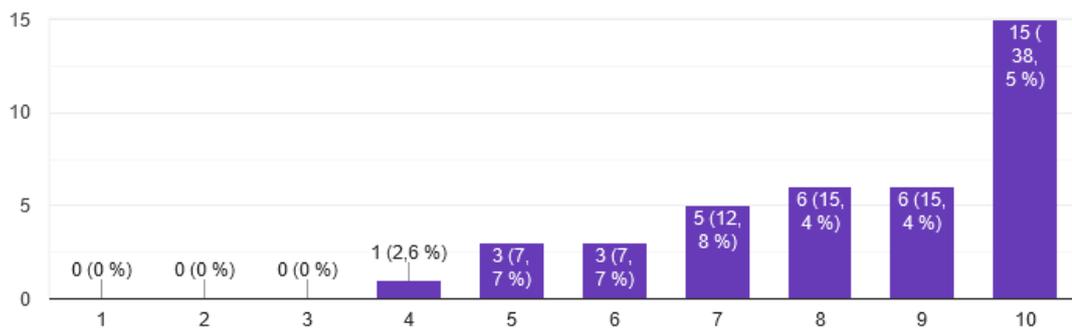


Figura 42. Puntuaciones de la curva de aprendizaje

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

39 respuestas

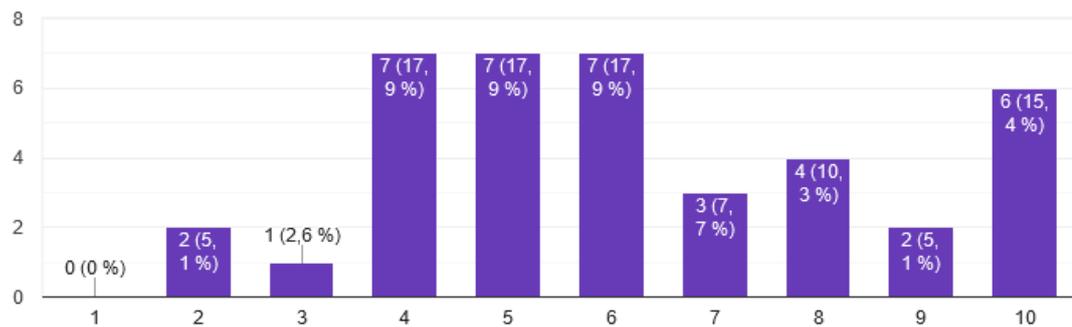


Figura 43. Puntuaciones sobre la dificultad del juego

En la Figura 42, siendo el uno “Muy difícil aprender a jugar” y el diez “Muy fácil aprender a jugar”, la mayoría de encuestados consideran que el juego es fácil de aprender, aunque se observan también algunas respuestas del orden de cuatro, cinco o seis que demuestran que para otros existen dificultades a la hora de interiorizar los controles. Estas respuestas se consideran

importantes dado que en el producto final al que se busca llegar es de vital importancia que los controles sean fáciles para todos los jugadores, no solo para la mayoría.

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

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

24 respuestas

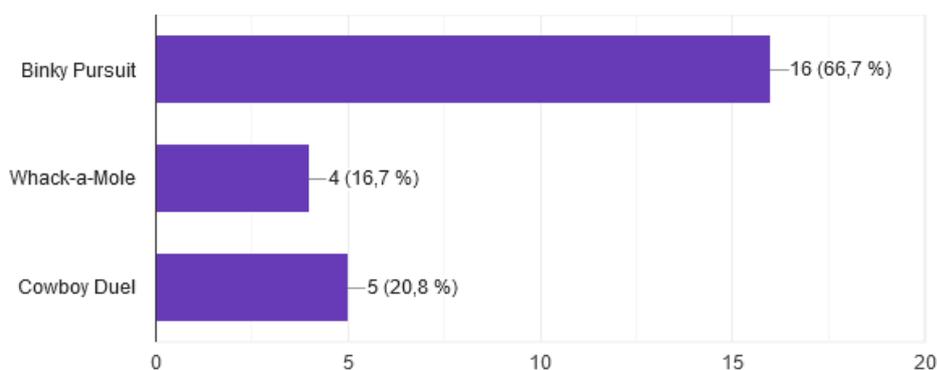


Figura 44. Opinión sobre dificultad excesiva de algún minijuego

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

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

39 respuestas

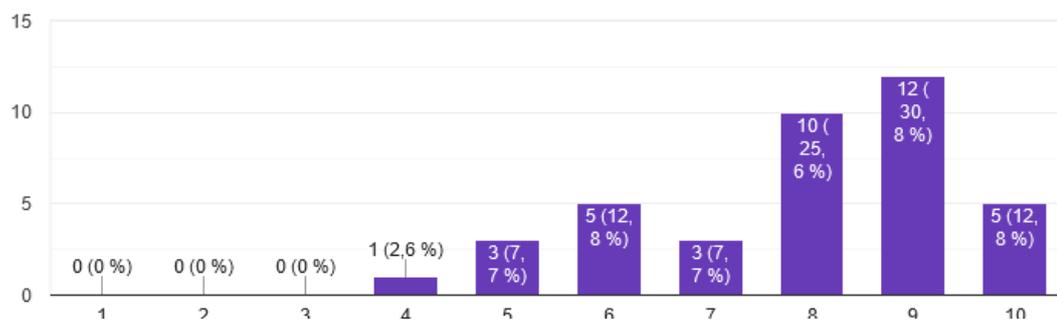


Figura 45. Opiniones sobre la claridad de la interfaz gráfica



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

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

39 respuestas

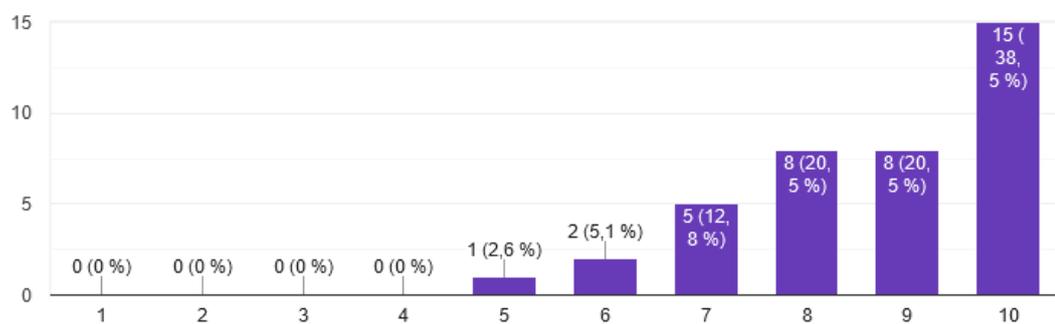


Figura 46. Opiniones sobre diseño y ambientación

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

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

39 respuestas

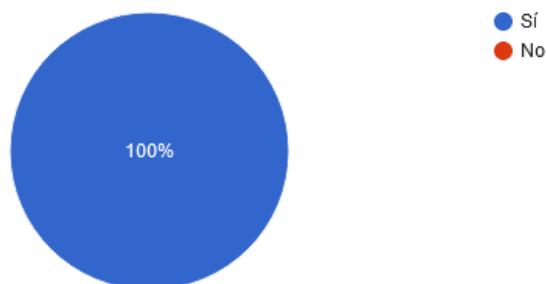


Figura 47. Opinión sobre la música de los minijuegos

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

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



39 respuestas

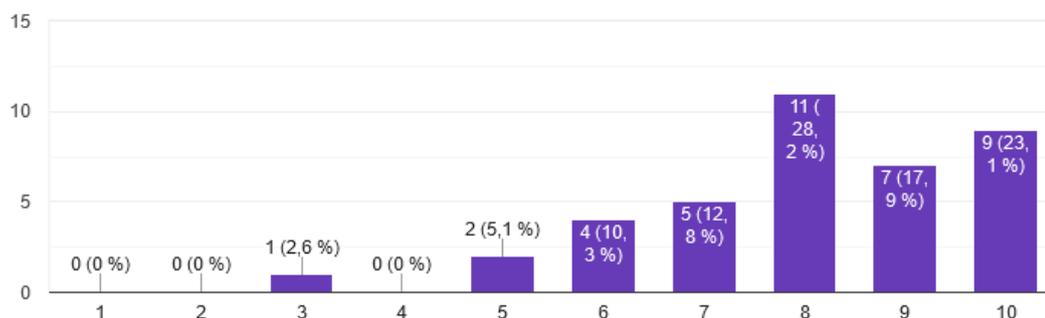


Figura 48. Puntuaciones generales de Blitz Party

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

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



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

3.3 Segundo *Minimum Viable Product*

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

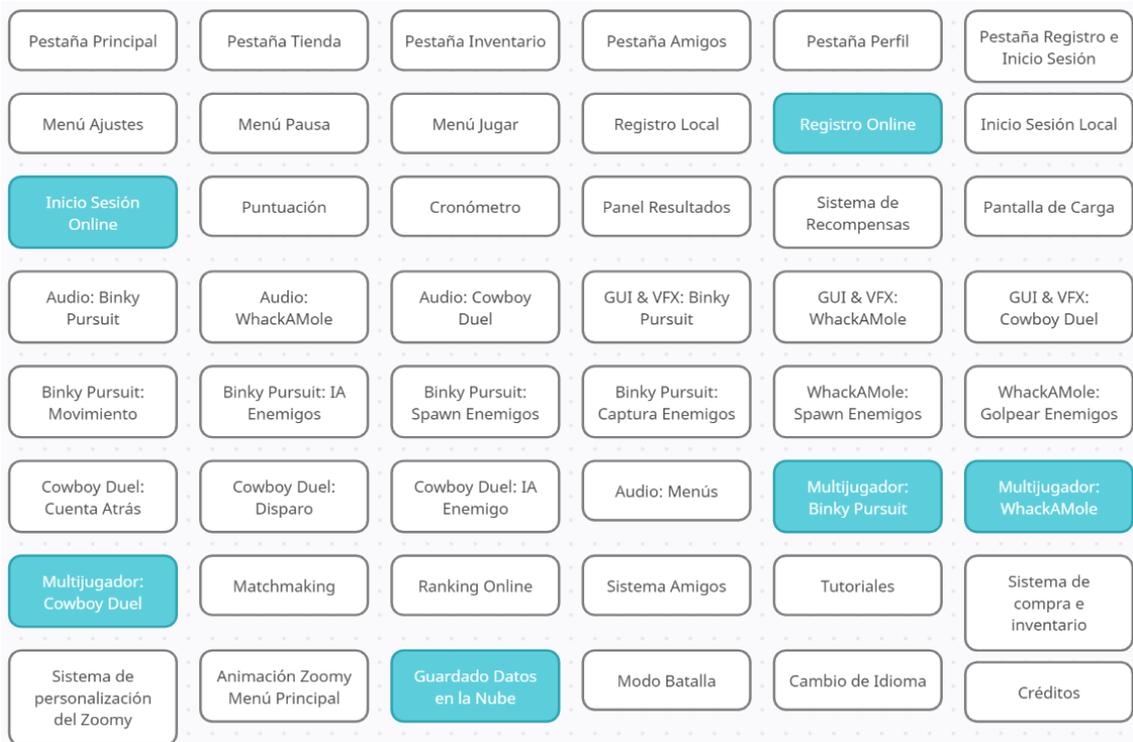


Figura 49. Mapa de características del segundo MVP

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

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

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

3.3.1 Desarrollo del MVP

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

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

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



Figura 50. Binky Pursuit multijugador

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

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



Figura 51. *Whack-A-Mole* multijugador

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

3.3.2 Experimento

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

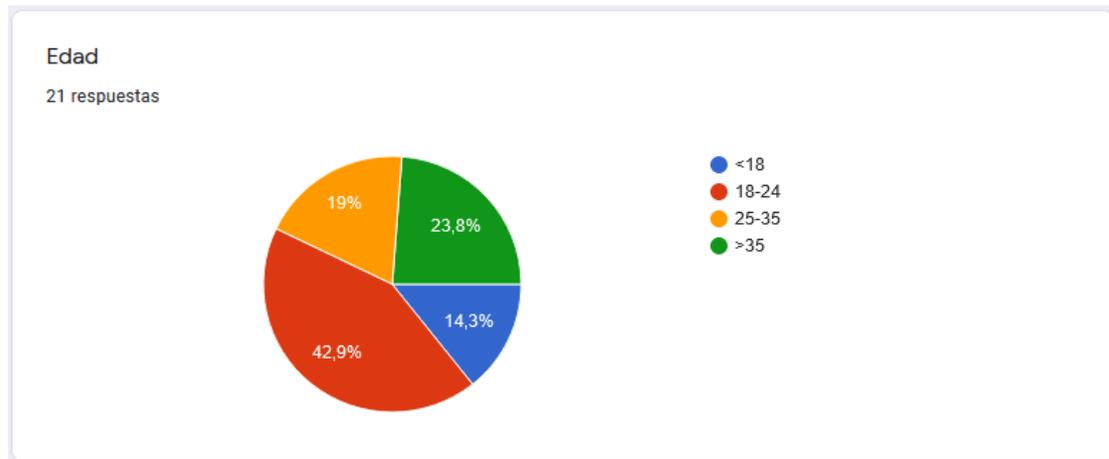


Figura 52. Pregunta rango de edad

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

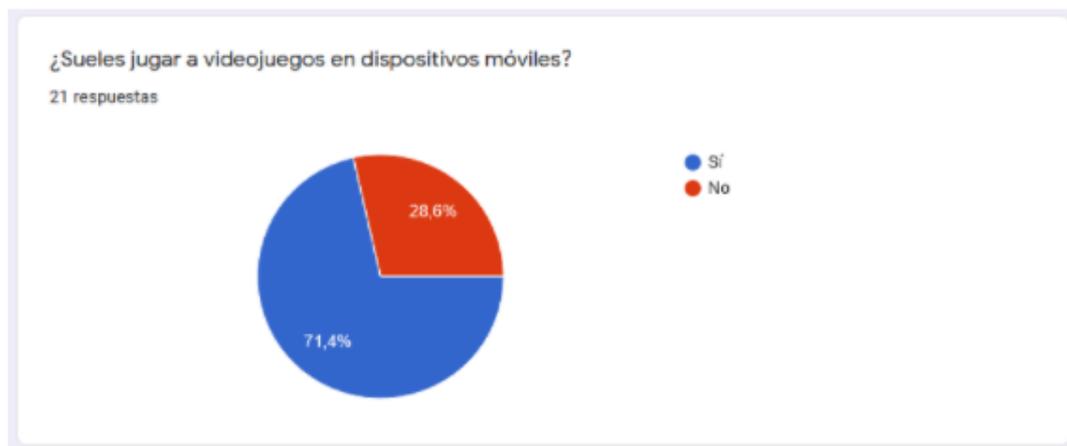


Figura 53. Pregunta jugadores de dispositivos móviles

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

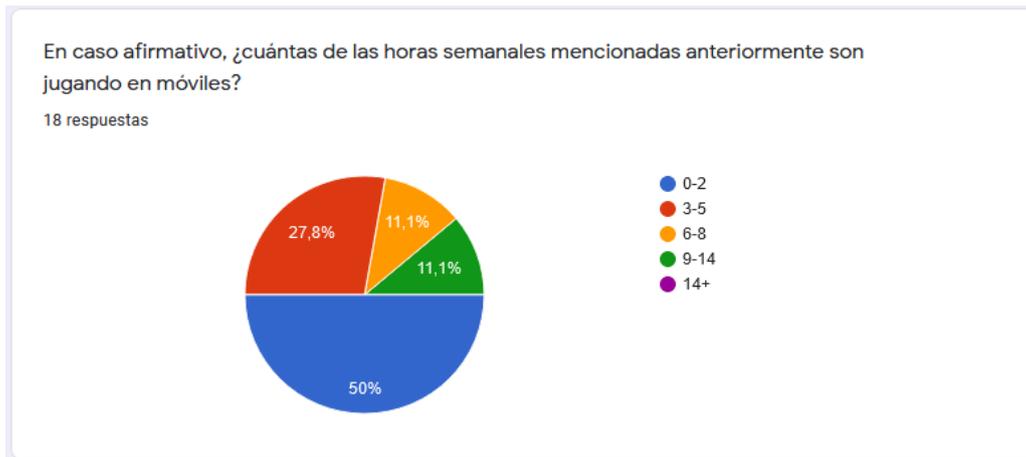


Figura 54. Pregunta horas jugadas en dispositivos móviles

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

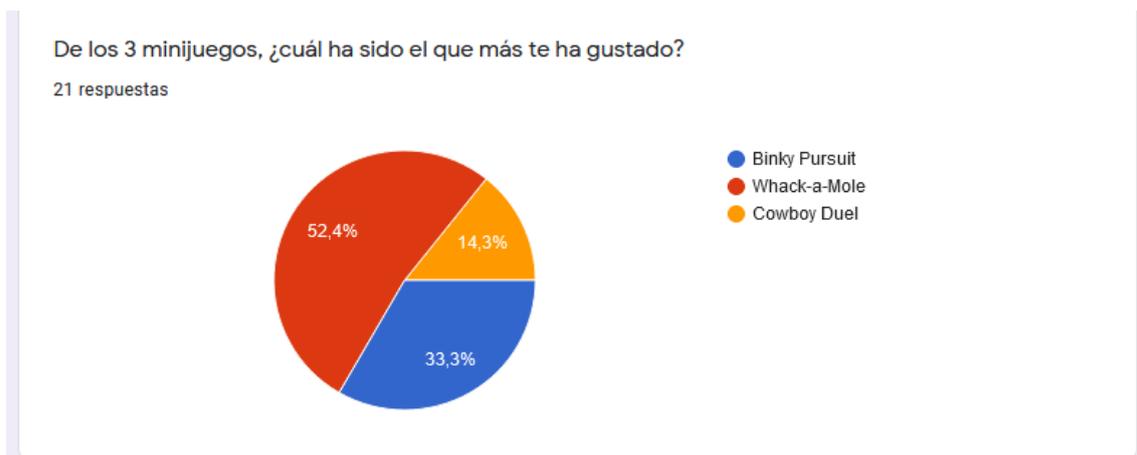


Figura 55. Pregunta minijuego preferido

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

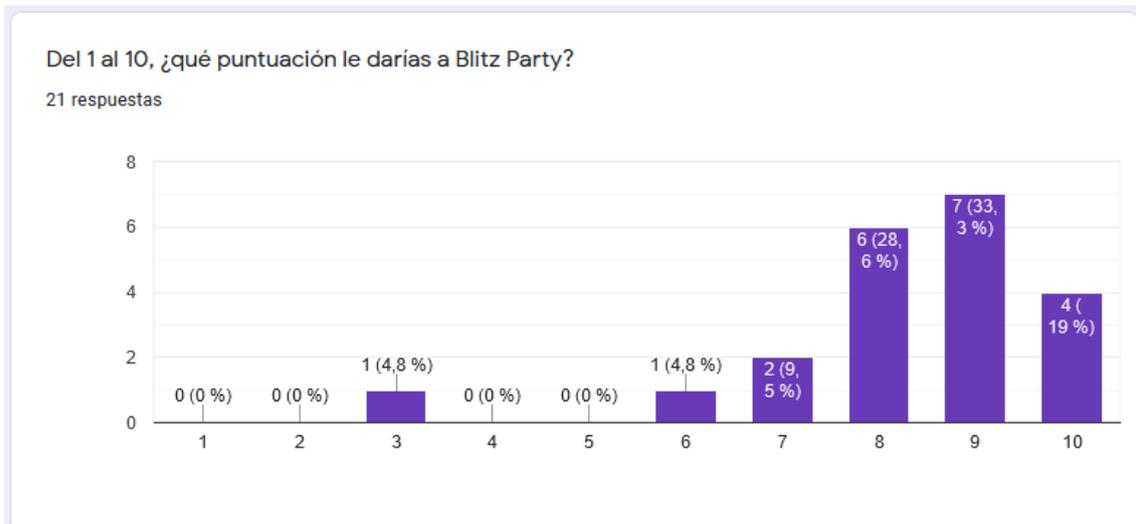


Figura 56. Pregunta puntuación general *Blitz Party*

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

3.4 Despliegue del producto y *marketing*

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

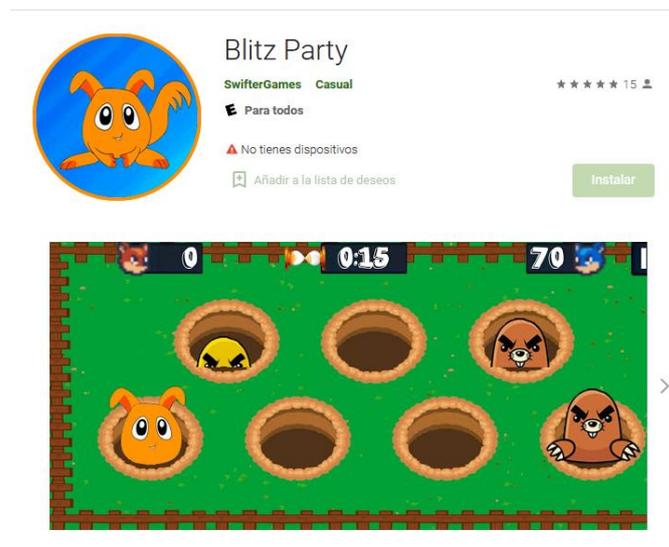


Figura 57. *Blitz Party* en la *Play Store*

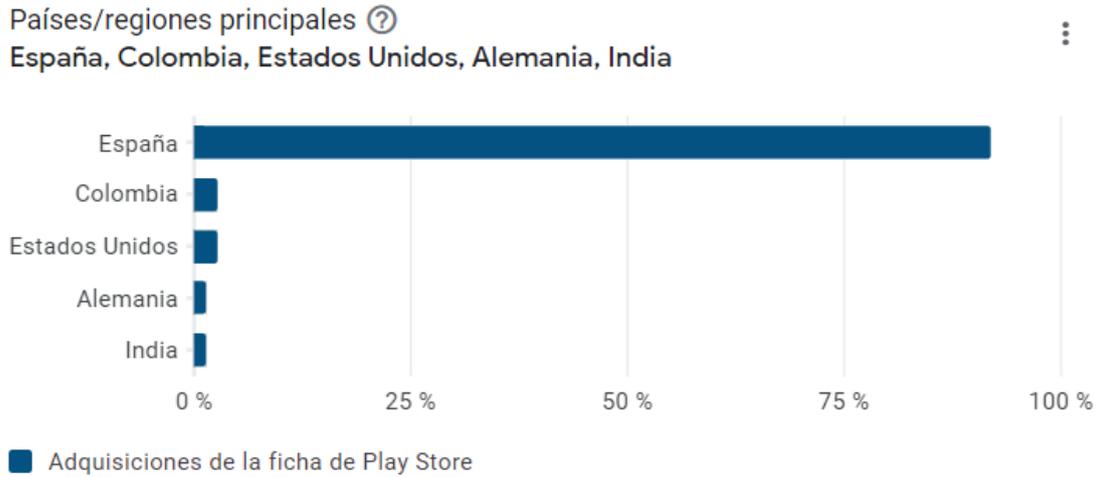


Figura 58. Países desde los que se ha descargado *Blitz Party*

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



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

4. Tecnologías utilizadas

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

4.1 Unity Engine



Figura 60. Logo de Unity Engine

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

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

4.2 Visual Studio



Figura 61. Logo del IDE Visual Studio

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

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

4.3 Git/GitHub



Figura 62. Logos de Git y GitHub respectivamente

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

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

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

4.4 Worki Process



Figura 63. Logo de Worki TUNE-UP Process

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

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

4.5 Photoshop



Figura 64. Logo de Adobe Photoshop

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

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

4.6 Mirror

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



Figura 65. Logo de Mirror

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

4.7 Azure PlayFab

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



Figura 66. Logo de *Azure PlayFab*

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

5. Análisis y diseño del multijugador

5.1 Análisis

5.1.1 Brainstorming

Al inicio del desarrollo del proyecto, se planteó y realizó un brainstorming para tener una visión clara del proyecto, por dónde dirigirlo y qué ideas nuevas podían surgir. Entre las ideas que surgieron (Figura 67) se encontraba el adaptar cada uno de los minijuegos que estuvieran hechos para entonces al multijugador, siendo estos duelos 1 vs 1 entre dos jugadores. Se barajó la posibilidad de crear un sistema online en un servidor dedicado o por el contrario implementarlo en red LAN. Debido a la complejidad del proyecto y la dificultad de pagar el costo de un servidor dedicado se optó por la opción más viable, hacer el multijugador para redes LAN.

En un primer momento, el equipo quería que el juego contuviera alguna especie de torneo online entre jugadores, así como un sistema de ranking global donde los jugadores pudieran mostrar su faceta más competitiva. Esto incluía o bien, la opción de crear un sistema de *lobbies* donde los jugadores se unen a las salas creadas por otros jugadores y cuando ambos estuvieran listos empezar la partida o bien, desarrollar un sistema de *matchmaking* haciendo uso de la herramienta *PlayFab* para emparejar a dos jugadores que estuvieran buscando partida en ese momento en cualquier parte del mundo. Se pensó que la forma más sencilla de proporcionar un sistema funcional y que no demorara el resto del desarrollo del proyecto era la de implementar un *matchmaking* para encontrar partidas *host* creadas por otros usuarios dentro de la misma red LAN y en caso de no existir, crearla para que otros pudieran unirse.



Figura 67. *Brainstorming* del multijugador

5.1.2 Cuestionario

Cuando se estaba planeando como diseñar e implementar el videojuego, se decidió realizar un cuestionario sobre aspectos generales para guiar al equipo hacia el buen camino. Con respecto al multijugador, existía una duda en uno de los minijuegos Whack-A-Mole, donde se barajaba la posibilidad de enfocarlo bien a partidas individuales donde al finalizar se comparaban las puntuaciones de ambos jugadores, o bien, el minijuego sucedía en la misma partida para ambos jugadores y el primero en golpear con el mazo al topo ganaba el punto.

Como se observa en la Figura 68, sobre la pregunta realizada a los encuestados, de los 17 participantes el 88,2% preferían jugar contra su rival en la misma partida compartiendo así los topes. Esto nos hizo ver que los encuestados preferían una faceta más competitiva del juego, donde la prioridad era hacer del multijugador un lugar de encuentro y entretenimiento.



Figura 68. Encuesta análisis multijugador del minijuego *Whack-A-Mole*

5.2 Diseño

5.2.1 Llamadas de Procedimiento Remotas

En cuanto al diseño, todos los minijuegos siguen la estructura del *framework* de *Mirror*, siguiendo la arquitectura cliente-servidor. Uno de los dos dispositivos es el *host*, el cual actúa como cliente local y servidor a la vez, mientras que el otro dispositivo se conecta como cliente remoto como se aprecia en la Figura 69.

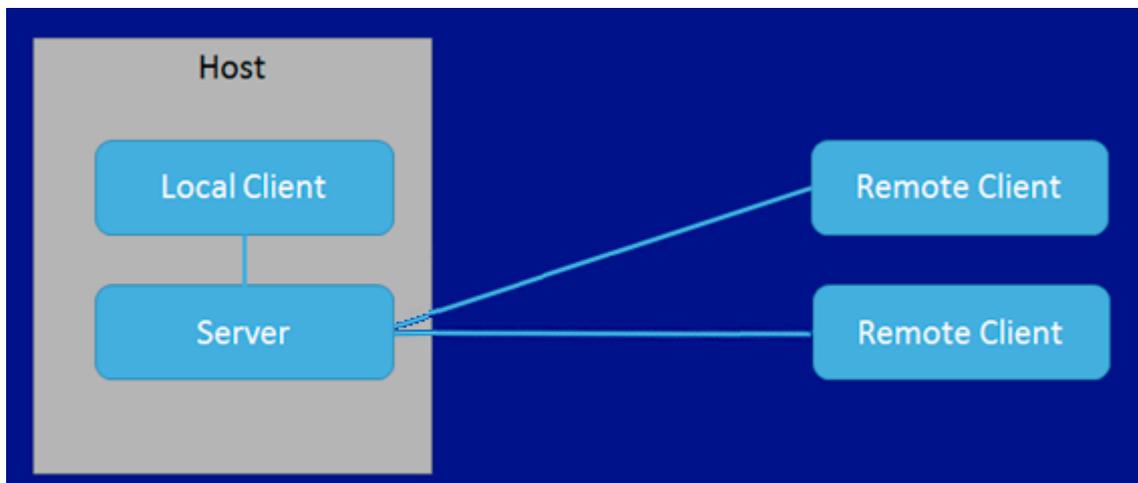


Figura 69. Funcionamiento de un dispositivo *Host*

En *Mirror* el servidor es una copia del proyecto de Unity, y siendo *host* este comparte el mismo proceso que el cliente local, por tanto, cada script contiene el código del servidor y del cliente. Debido a esto, las clases se han diseñado siguiendo este patrón, por lo que internamente los *scripts* comparten mensajes para pasar información entre clientes y servidor. El funcionamiento de este enfoque es el de comunicarse mediante *Remote Procedure Call (RPC)*, donde el cliente debe

ejecutar un tipo de método específico para comunicarse con el servidor y solicitar alguna acción, y este contesta ejecutando código en el lado de los clientes.

Como se observa en la Figura 70, la comunicación entre clientes y servidor se realiza mediante el uso de comandos, *Commands* dentro de *Mirror*, los cuales se ejecutan en el lado del cliente, pero internamente se gestiona por el *framework*, enviando los mensajes necesarios al servidor con toda la información para poder ejecutar el código especificado. El objeto que tiene adherido el *script* debe tener autoridad dada por el servidor para ser capaz de comunicarse con él, aunque *Mirror* facilita una opción para comunicarse sin necesidad de tener autoridad.

El servidor puede comunicarse con todos los clientes o con el que quiera por separado, por tanto, cuando el servidor quiere que un cliente realice una acción o simplemente actualizar el estado de sus datos, puede hacerlo de dos formas. La primera, mediante el uso de *ClientRPC*, que es un tipo de llamada remota que se ejecutará en el lado del cliente, en cada uno de los clientes incluyendo el local al *host*.

Por otro lado, si solo se quiere enviar la ejecución del método a un único cliente, se hará por medio de *TargetRPC*. Este tipo de llamada remota se ejecutará en el cliente especificado, indicando la conexión de dicho cliente en el primer parámetro del método y en caso no de indicarse, se ejecutará en el cliente al que pertenezca el objeto con dicho *script*, es decir, el cliente con autoridad que solicitó al servidor ejecutar ese RPC.

El servidor, al compartir código con el cliente local, conoce las variables del *script* desde el que se le llama, pero si el cliente necesitara pasarle información, cualquier método marcado como *Command* admite parámetros, que el servidor recibirá y podrá utilizar. Estos parámetros, deben ser de algún tipo soportado por *Mirror*, descritos a continuación:

- Tipos primitivos de C# (*byte*, *int*, *char*, *uint*, *UInt64*, *float*, *string*, etc)
- Tipos incluidos en *Unity* (*Vector3*, *Quaternion*, *Rect*, *Plane*, *Vector3Int*, etc)
- Objeto de tipo URI
- Objeto del tipo *NetworkIdentity*
- *GameObjects* con un componente *NetworkIdentity* adjunto
- *Structs* con cualquier tipo de los mencionados anteriormente
- Clases siempre que contengan campos de tipos soportados.
- *ScriptableObjects* siempre que sus campos sean de tipos soportados.
- *Arrays* de los tipos mencionados anteriormente.
- *ArraySegments* de los tipos mencionados anteriormente.



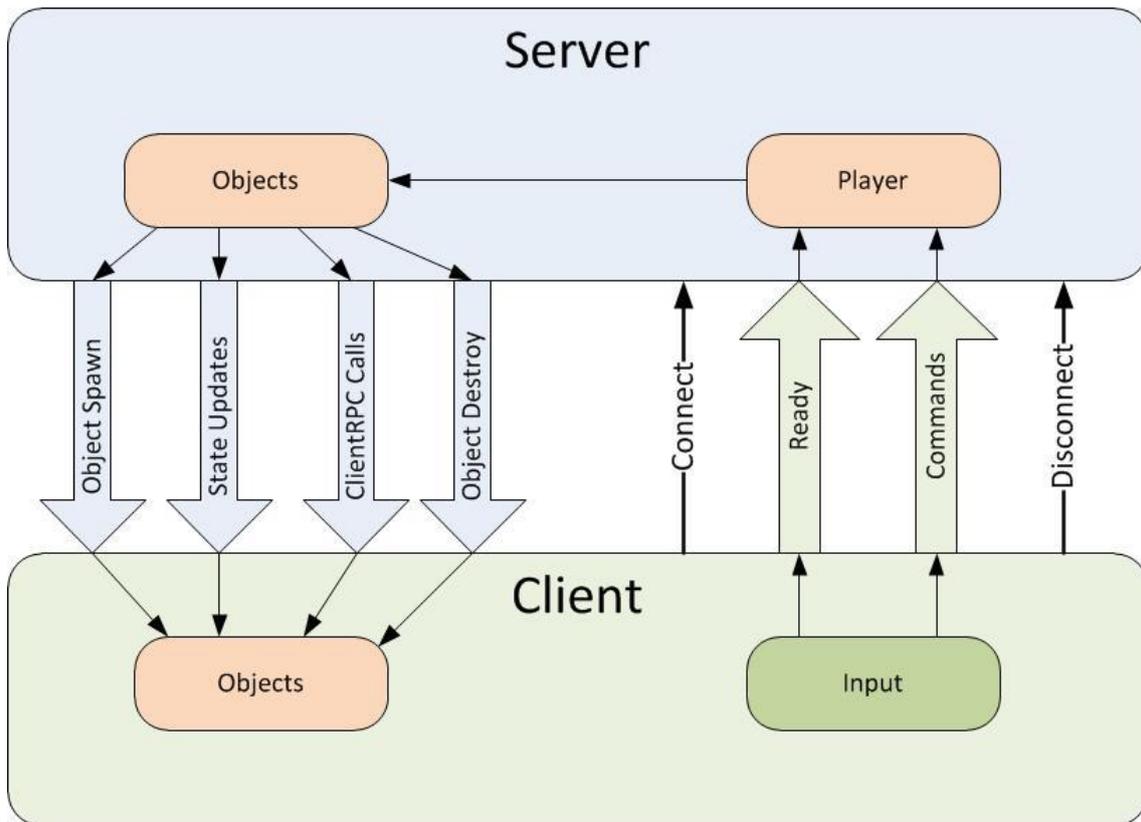


Figura 70. Funcionamiento comunicaciones remotas entre cliente y servidor en *Mirror*

5.2.2 Comunicaciones entre scripts

Binky Pursuit

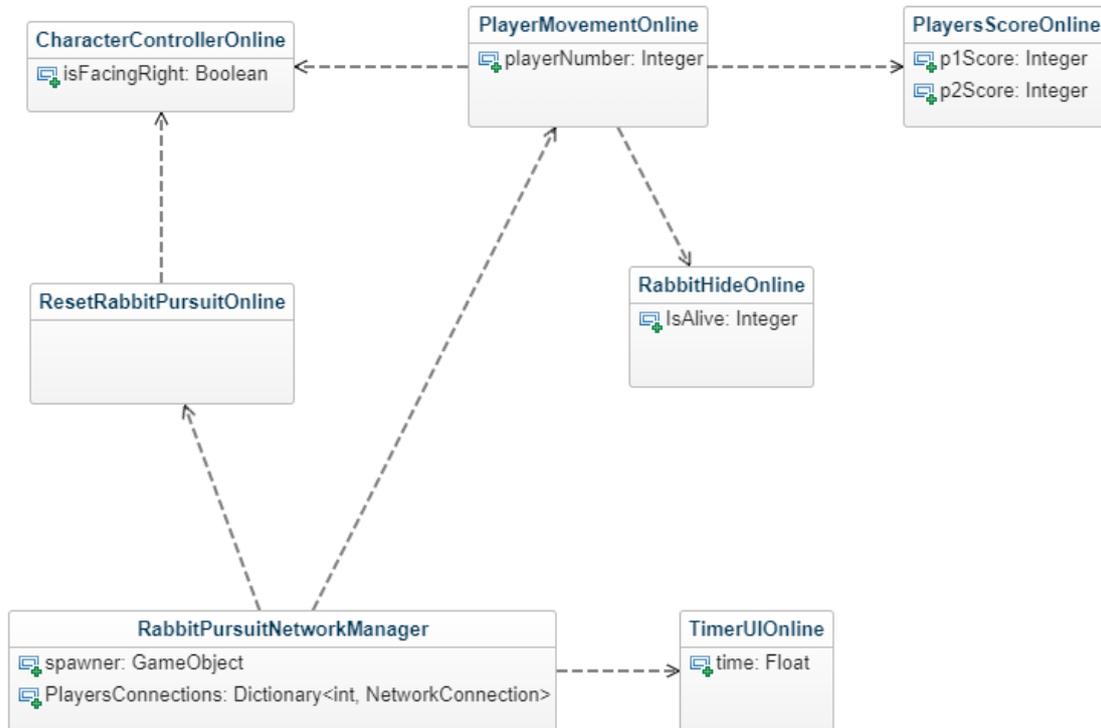


Figura 71. Diagrama de dominio de los scripts de *Binky Pursuit*

La Figura 71, muestra las relaciones entre los *scripts* y cómo se comunican para hacer funcionar el minijuego. El más importante, *RabbitPursuitNetworkManager*, se encarga no solo de crear el *host* y gestionar la conexión del otro cliente, sino inicializar aspectos clave para el funcionamiento del juego. Es por ello por lo que hace uso de *TimerUIOnline*, *ResetRabbitPursuitOnline* y *PlayerMovementOnline* para preparar el cronómetro, el controlador de movimiento del jugador y su número identificativo.

PlayerMovementOnline es el *script* principal del jugador y este se encarga tanto del control de la posición y movimiento haciendo uso de *CharacterControllerOnline*, como de sumar puntos al jugador si consigue golpear un enemigo con *PlayersScoreOnline*. *RabbitHideOnline* informa al jugador de si el enemigo sigue vivo, de esta forma, si hubiera problemas de red, el servidor no intentaría eliminar dos o más veces la misma referencia del enemigo cuando este es atrapado por el jugador.

Whack-A-Mole

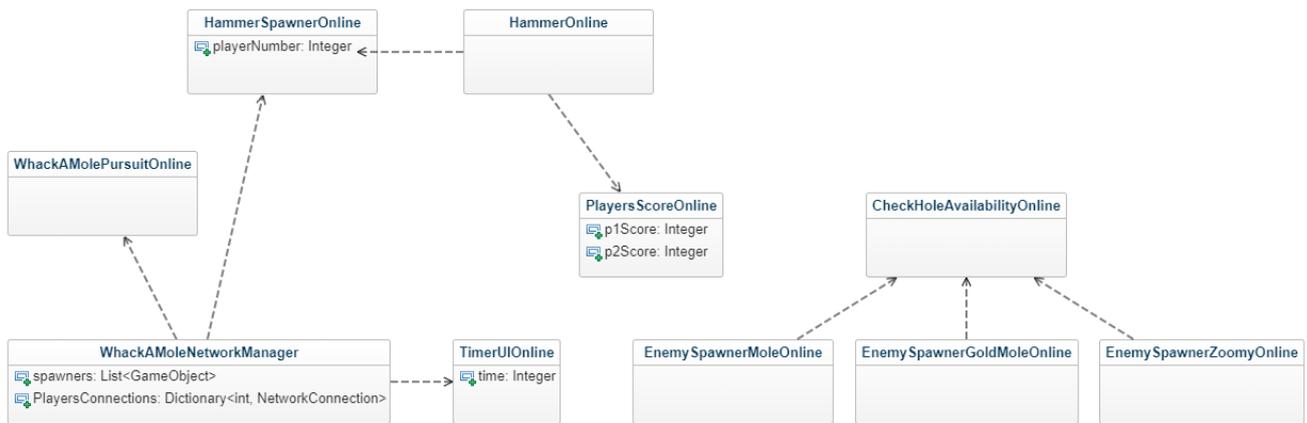


Figura 72. Diagrama de dominio de los *scripts* de *Whack-A-Mole*

Como se observa en la Figura 72, *WhackAMoleNetworkManager* inicializa el cronómetro, y se encarga de tocar aspectos generales del minijuego como la creación de los tres tipos de *spawner* y la gestión de conexiones de clientes. Cada uno de los *spawners* hace uso de *CheckHoleAvailabilityOnline* para saber qué puntos de creación se están utilizando en ese momento. Si un hueco está ocupado por un objeto instanciado, entonces el *spawner* busca otro disponible para evitar conflictos.

El *script* principal del jugador, *HammerSpawnerOnline*, almacena el identificador del jugador e instancia los mazos con el *script* *HammerOnline*, el cual recibe una referencia del jugador que lo crea para posteriormente poder sumar puntos a dicho jugador con *PlayersScoreOnline*.

Cowboy Duel

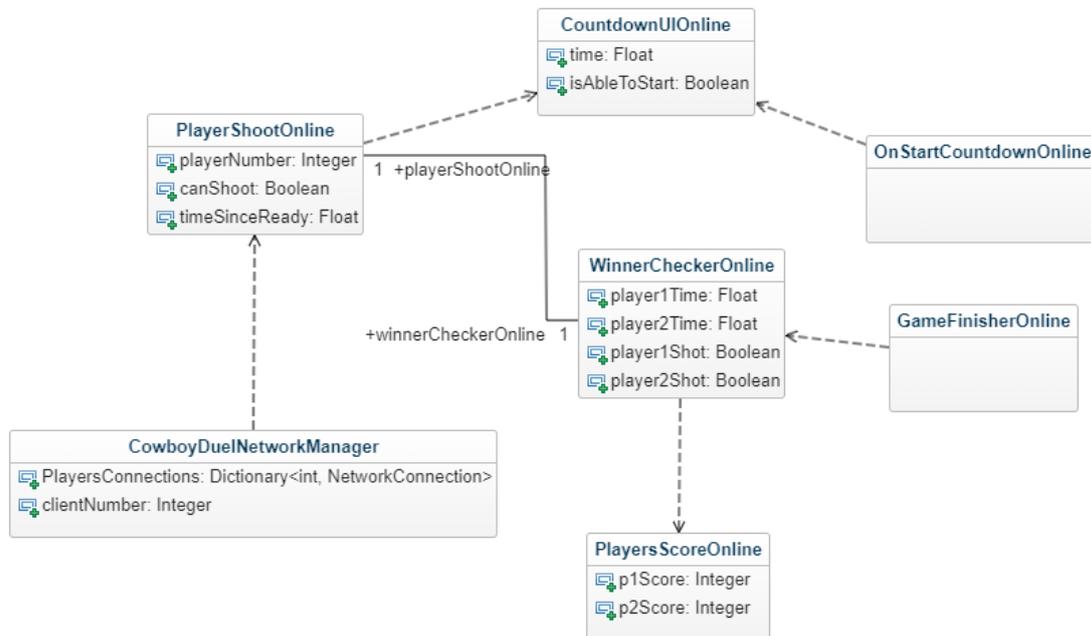


Figura 73. Diagrama de dominio de los *scripts* de *Cowboy Duel*

En la Figura 73, el *script* *CowboyDuelNetworkManager* gestiona las conexiones de los clientes, además de habilitar las animaciones mediante una referencia al *script* de *PlayerShootOnline*. Este último *script*, se encarga de gestionar las acciones de los usuarios y sincronizarlas con el cronómetro de *CountdownUIOnline*, así como enviar información a *WinnerCheckerOnline* que necesita como el tiempo desde que puede disparar hasta que lo hace, la cual utiliza para dictaminar que jugador ha ganado la ronda. *WinnerCheckerOnline* necesita una referencia a *PlayersScoreOnline* para sumar un punto al correspondiente ganador de la ronda.

OnStartCountdownOnline, inicializa el cronómetro de *CountdownUIOnline* mediante una referencia directa. Por último, *GameFinisherOnline* se encarga de notificar la finalización del minijuego una vez uno de los jugadores ha alcanzado dos puntos.

6. Mirror

6.1 Introducción

En esta sección se detallan los pasos realizados y herramientas utilizadas para proporcionar al juego la funcionalidad multijugador y permitir la interacción de dos jugadores en cada uno de los minijuegos.

La organización del aspecto *online* sigue el mismo esquema que su contraparte *offline*. Se ha adaptado cada uno de los minijuegos en su versión local para tener una versión que permite partidas entre jugadores. Se ha añadido un apartado para jugar en LAN dentro de una misma red.

A continuación, se presentan los puntos en los que se divide esta parte, empezando por una explicación y funcionamiento de Mirror, seguido de cómo se han desarrollado los diferentes minijuegos utilizando esta herramienta.

6.2 Mirror Framework

La parte multijugador hace uso del *framework* de *Mirror* por su modelo cliente-servidor, que permite una versatilidad enorme a la hora de estructurar el juego ya sea bien para utilizar un servidor dedicado y cada dispositivo actuar como cliente o bien uno de los dos dispositivos hace de host y el otro se conecta a este.

La instalación de la herramienta no tiene complicación alguna debido a que se encuentra disponible en la *Asset Store* de *Unity*, o bien en caso de querer probar una versión más nueva en fase beta, se puede conseguir de su propia página de GitHub y descargarse el proyecto.

Mirror hace uso de componentes que se añaden a los objetos de Unity para dar funcionalidad al juego. Cada uno de estos componentes proporcionan características básicas y plenamente funcionales, pero el usuario puede heredar de estos componentes y adaptarlos a sus necesidades si la funcionalidad base no cumple sus especificaciones y necesita personalizarlos para su juego. A continuación, se detallan características de Mirror y componentes utilizados en el proyecto:

6.2.1 Network Manager

El componente *Network Manager* controla los aspectos generales del multijugador como son la gestión del estado del juego, cambio de escenas dentro de *Unity*, el *spawn* de objetos dentro del mundo o información de rendimiento entre otros.

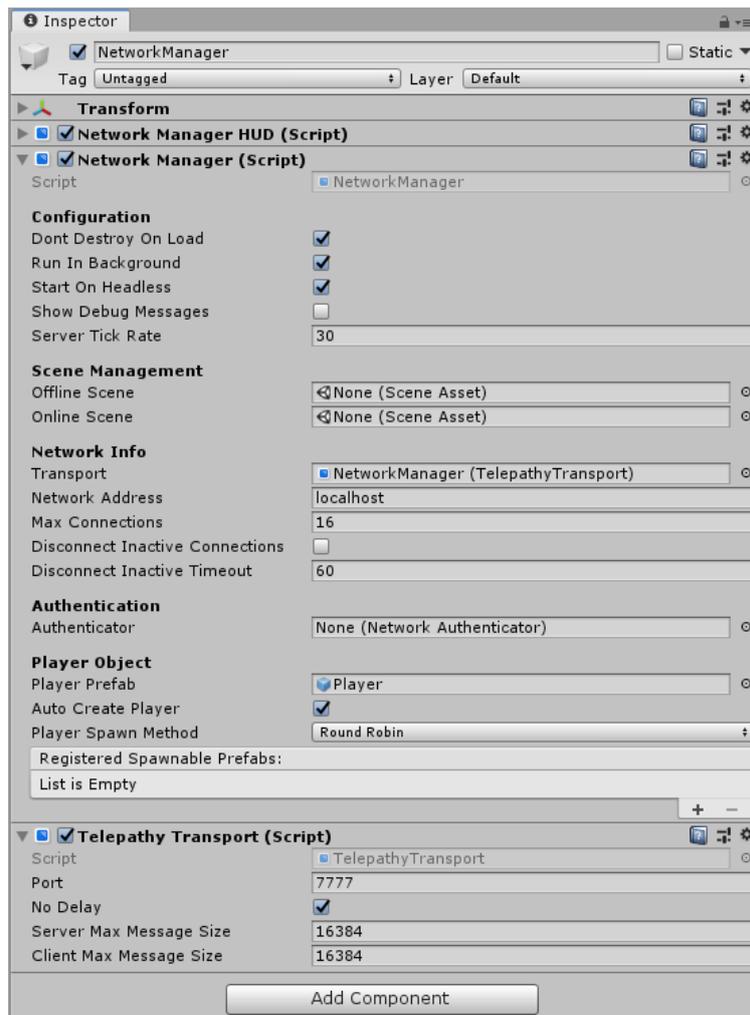


Figura 74. Componente *Network Manager* de *Mirror*

Como se observa en la Figura 74, este componente gestiona la dirección de red donde deben conectarse los clientes, así como su puerto, además del número máximo de conexiones que acepta. Permite personalizar variables como el tipo de objeto que se *spawna* por defecto al conectarse un cliente, el orden de aparición de estos o un listado de objetos para crear en tiempo de ejecución a través de la red y que puedan ser vistos por todos los clientes conectados. El componente está diseñado para poder crearse una *build* del juego *headless* para ser alojada en un servidor dedicado, así como establecer una frecuencia de actualización para adaptar el funcionamiento al género de videojuego.

Network Manager se ha construido teniendo en cuenta el uso de diferentes protocolos de comunicaciones. Entre los protocolos diseñados por *Mirror* destacan:

- *KCP Transport*: Este protocolo viene por defecto cuando se añade un *Network Manager* a algún objeto. Está basado en *kcp.c* y se ha migrado completamente a C#, siendo altamente eficiente y sencillo, además de haber sido sometido a numerosas pruebas.

- *Telephaty Transport*: Protocolo diseñado para juegos del género MMO los cuales escalan con un número grande de usuarios. Es fiable y extremadamente eficiente debido a su funcionamiento mediante un sistema de mensajes simples TCP.
- *WebSockets Transport*: *WebSockets* es un protocolo diseñado para la plataforma WebGL en Unity. Esta plataforma tiene su propio protocolo porque muestra incompatibilidad con los demás protocolos.
- *Multiplex Transport*: No se considera un protocolo como tal porque su función real es la de permitir tener varios protocolos trabajando juntos. Se puede hacer uso de *Multiplex* para combinar, por ejemplo, el protocolo *Telephaty* usando TCP con el protocolo *WebSockets* para WebGL y plataformas web, permitiendo así las interacciones entre usuarios de navegador y usuarios de otras plataformas como PC o consolas.

Mirror permite añadir un componente llamado *Network Manager HUD* que ofrece una interfaz donde el usuario puede iniciar su instancia del juego como cliente, servidor o host y hacer de ambos, facilitando probar las funcionalidades implementadas rápidamente.

6.2.2 Network Discovery

Como se ha explicado en el apartado de arriba, *Network Manager* necesita una dirección de red y un puerto para que los clientes puedan conectarse al servidor. Esto en servidores dedicados es su funcionalidad base, ya que se encuentran en un sitio concreto y su IP no varía y es conocida, pero para conexiones dentro de una misma LAN, diferentes dispositivos que actúen de anfitrión tendrán diferentes IPs. Esto es innecesariamente complejo para los usuarios, teniendo que averiguar la IP del dispositivo al que se conectan.

Para esto *Mirror* ofrece el componente *Network Discovery*, mostrado en la Figura 75, que realiza una búsqueda intensiva de instancias de juego que actúen como servidor solo o anfitrión dentro una misma red LAN y permite al usuario conectarse a las partidas que aparezcan en el listado.

Para permitir esta interacción visual, se añade un componente homólogo a *Network Manager HUD*, llamado *Network Discovery HUD*, que al igual que el anterior, proporciona una interfaz al usuario para crear u hospedar un servidor o bien para buscar una partida siendo cliente.

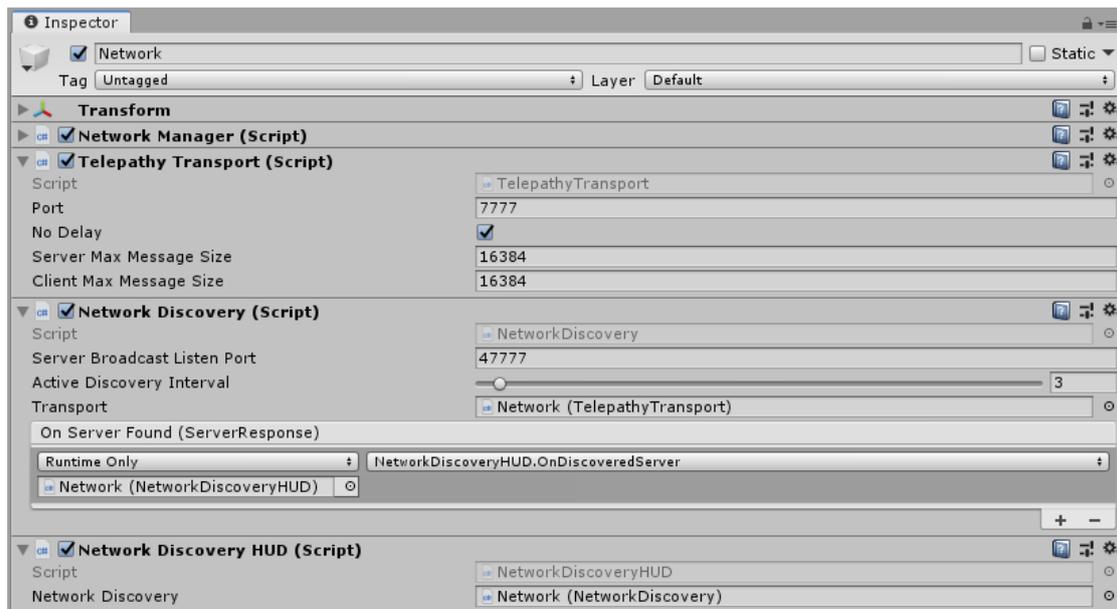


Figura 75. Componente *Network Discovery* de *Mirror*

6.2.3 *Network Identity*

Network Identity es el componente principal y más importante del ecosistema de *Mirror*. Este componente sirve para identificar cada objeto dentro de la red. Para poder *spawnear* un objeto en la red, estos deben tener obligatoriamente un *Network Identity* para que el entorno les asigne un identificador único. Esto se aplica tanto para objetos creados cuando carga una escena como para objetos creados en tiempo de ejecución. En caso de no incluirse se muestran errores de compilación cuando se asocia el objeto al *NetworkManager* o de ejecución cuando se intenta crear un objeto en red sin el componente.

Mediante estos identificadores, el *framework* es consciente y conoce cada uno de los objetos que van a interactuar de alguna manera en la red. Además, ofrece la posibilidad de que algunos objetos solo existan en el lado del servidor para que sean exclusivamente gestionados por este, mediante una casilla que puede ser marcada tal como se ve en la Figura 76.

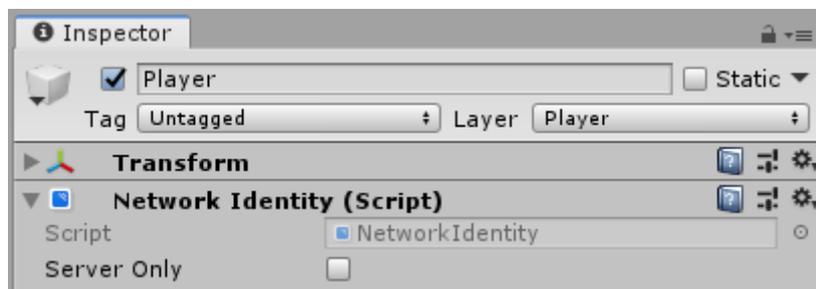


Figura 76. Componente *Network Identity* de *Mirror*



6.2.4 Network Transform

Este componente se utiliza para sincronizar la posición, la rotación y la escala de los objetos dentro de la red. Cuando un objeto se mueve, rota o escala por código, el cambio solo es visible localmente dentro del cliente promotor de la acción. Para que se vea reflejado en los demás clientes, es necesario añadir el componente *Network Transform* el cual sincroniza el componente *Transform* de *Unity* encargado de gestionar la posición, rotación y escala de un objeto.

Cabe recalcar que *Mirror* añade un *Network Identity* al añadir un *Network Transform* si el primero no ha sido añadido con anterioridad, ya que precisa de uno para su funcionamiento. Si se precisa del cliente para mover el objeto en vez del servidor, se puede marcar la casilla de *Client Authority* (Figura 77) para dar autoridad al cliente, ya que por defecto es el servidor quien tiene la autoridad para hacerlo.

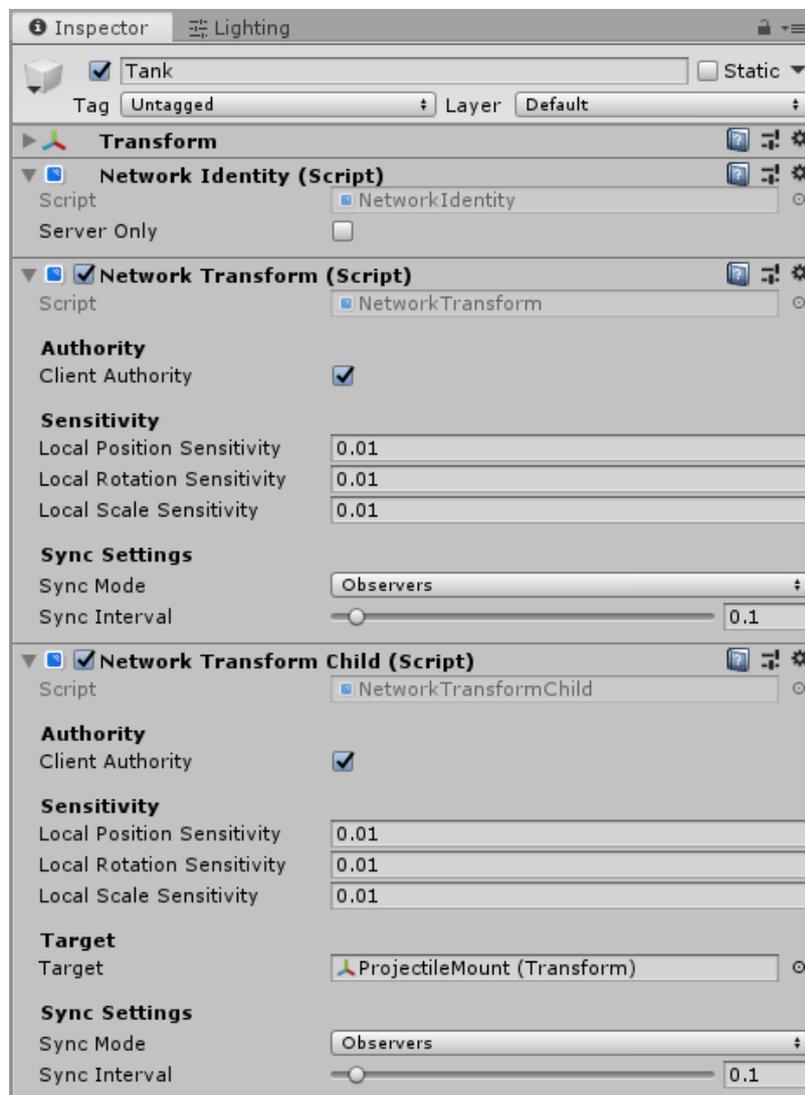


Figura 77. Componente *Network Transform* de *Mirror*

6.2.4 Network Rigidbody

Similar a *Network Transform*, este componente es el encargado de sincronizar todo lo relacionado a físicas de Unity y sus interacciones con los *Rigidbody* como pueden ser la velocidad y la gravedad.

Como también pasa con el componente mencionado anteriormente, el *Network Rigidbody* necesita un *Network Identity* para su funcionamiento y este será añadido en caso de no estar previamente. Además, se puede añadir el *Network Transform* para que trabaje junto al *Network Rigidbody* para que, mientras uno se encarga de la posición, el otro se encarga de la velocidad y la gravedad, produciendo un movimiento suave y sincronizado entre los clientes.

Como pasa con el otro componente homólogo, este puede dar autoridad al cliente para interactuar con las físicas en vez de hacerlo el servidor mediante la casilla de *Client Authority* como se enseña en la Figura 78.

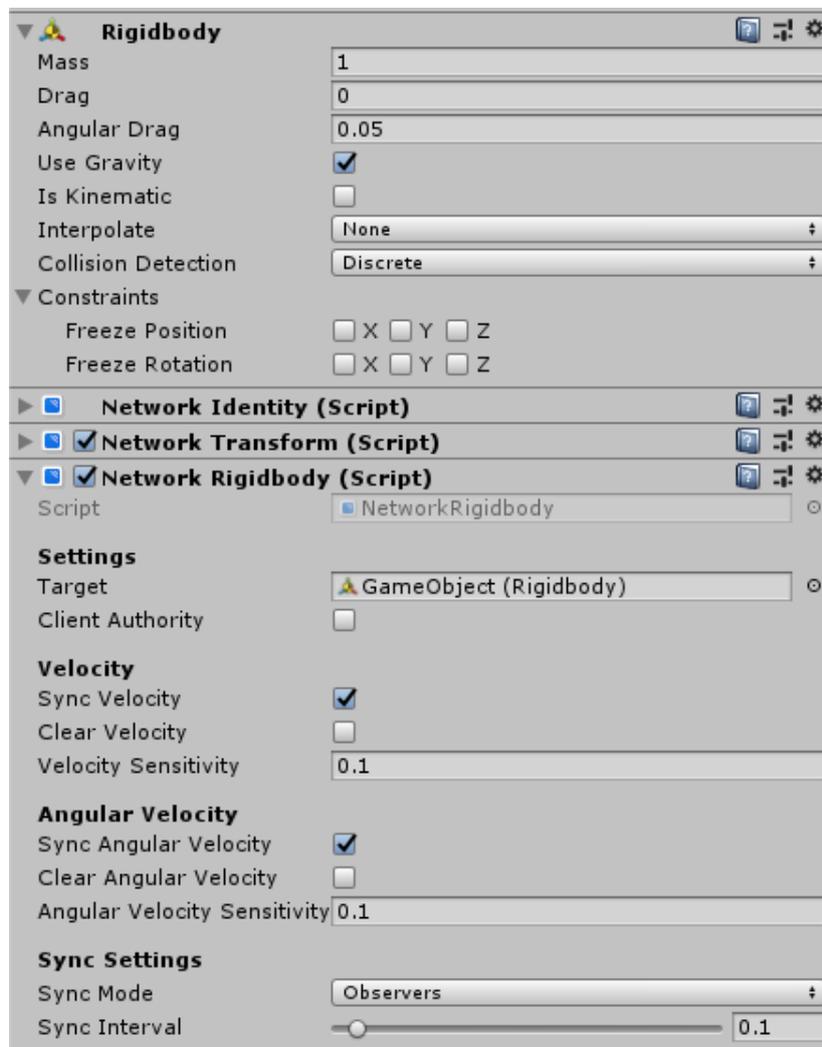


Figura 78. Componente *Network Rigidbody* de *Mirror*

6.2.4 *Network Animator*

Permite la sincronización de estados de animaciones entre componentes en red. Trabaja junto al componente Animator de Unity y su animator controller asociado para coordinar los estados y sus parámetros.

Las animaciones pueden realizarse localmente si el cliente tiene autoridad para ello, y el cliente lo retransmite a los otros clientes o bien, animar directamente el objeto en el servidor y este actualiza el estado y la información en los clientes. Para el caso de animar el cliente el objeto y notificar a través de la red al resto es necesario marcar la casilla de *Client Authority* (Figura 79).

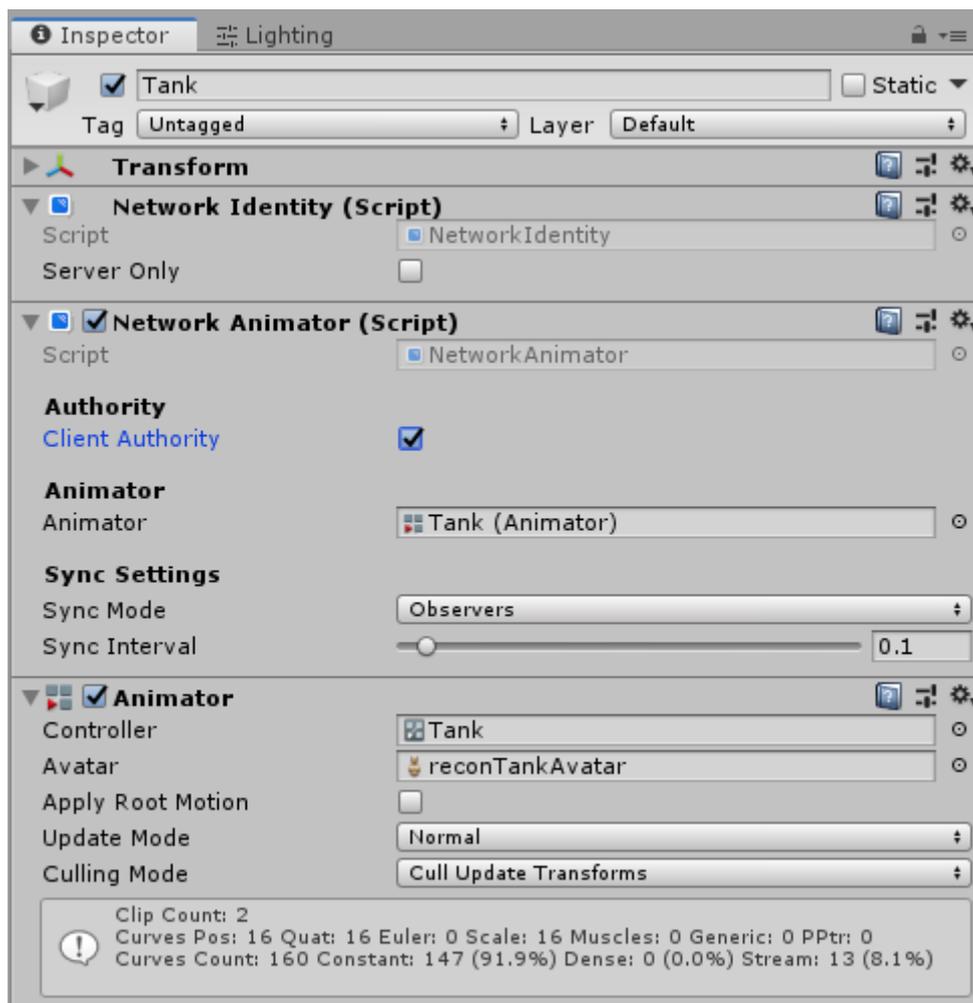


Figura 79. Componente *Network Animator* de *Mirror*

7. Implementación del multijugador

En cuanto a la implementación, se ha optado por adaptar cada minijuego a su versión multijugador sin estar relacionados y siendo independiente a la hora de jugar a cada uno de ellos. Para el desarrollo de los minijuegos, se parte de la base de sus homólogos en su versión de jugador en solitario, teniendo escenas propias para sus versiones online, pero manteniendo la base y la funcionalidad principal del código y haciendo modificaciones sobre este para adaptarlo.

7.1 Minijuegos

En los siguientes apartados sobre los distintos minijuegos, se introducen los aspectos clave de cada uno de ellos. Todos ellos son necesarios y de vital importancia a la hora de conseguir un correcto funcionamiento dentro del marco del framework. Se debe recalcar que todos los minijuegos utilizan el protocolo de transporte KCP, ofrecido por el *framework* de *Mirror*.

7.1.1 *Binky Pursuit*

NetworkManager

Como también pasa con los otros minijuegos, cada uno tiene su propio *Network Manager* que gestiona de forma personalizada los aspectos importantes del mismo y otros comunes como puede ser la limitación a dos jugadores por partida o el comienzo de esta una vez los dos jugadores se hayan conectado a la misma, como se había estipulado en su diseño. Además de lo mencionado anteriormente, se ha añadido un componente de tipo script llamado *NetworkLobbyController* que gestiona la conexión del cliente al *host* en caso de existir uno, haciendo una búsqueda en la misma LAN, y en caso contrario creando el mismo el *host* para que otro cliente remoto pueda unirse.

En este caso, el componente (Figura 80) se encarga de instanciar los dos clientes a través de la red para su visibilidad en ambos dispositivos y darles autoridad para poder hacer solicitudes al servidor y poder enviar comandos que este pueda ejecutar. Una vez creados y conectados los clientes, el *Network Manager* se encarga de deshabilitar el panel de búsqueda de usuarios y habilitar todos los elementos de la interfaz que permiten al jugador realizar sus acciones dentro del minijuego. A su vez, debe iniciar una animación que indique al usuario cuál es su personaje, siendo este indicador inequívoco en cada dispositivo. Cuando ha acabado toda la fase de preparación, el *manager* se encarga de inicializar y arrancar el cronómetro de tiempo y crear los diferentes *spawners* que se encargan de instanciar los enemigos a través de la red.

Cuando la partida finaliza, el componente tiene una serie de tareas a realizar como destruir todos los *spawners* y enemigos en la partida, además de hacer una limpieza de conexiones antes de cargar la siguiente escena para desconectar y destruir los objetos cliente de la partida, así como



desconectar y cerrar el *socket* del servidor para no tener problemas e interferir con los otros minijuegos.

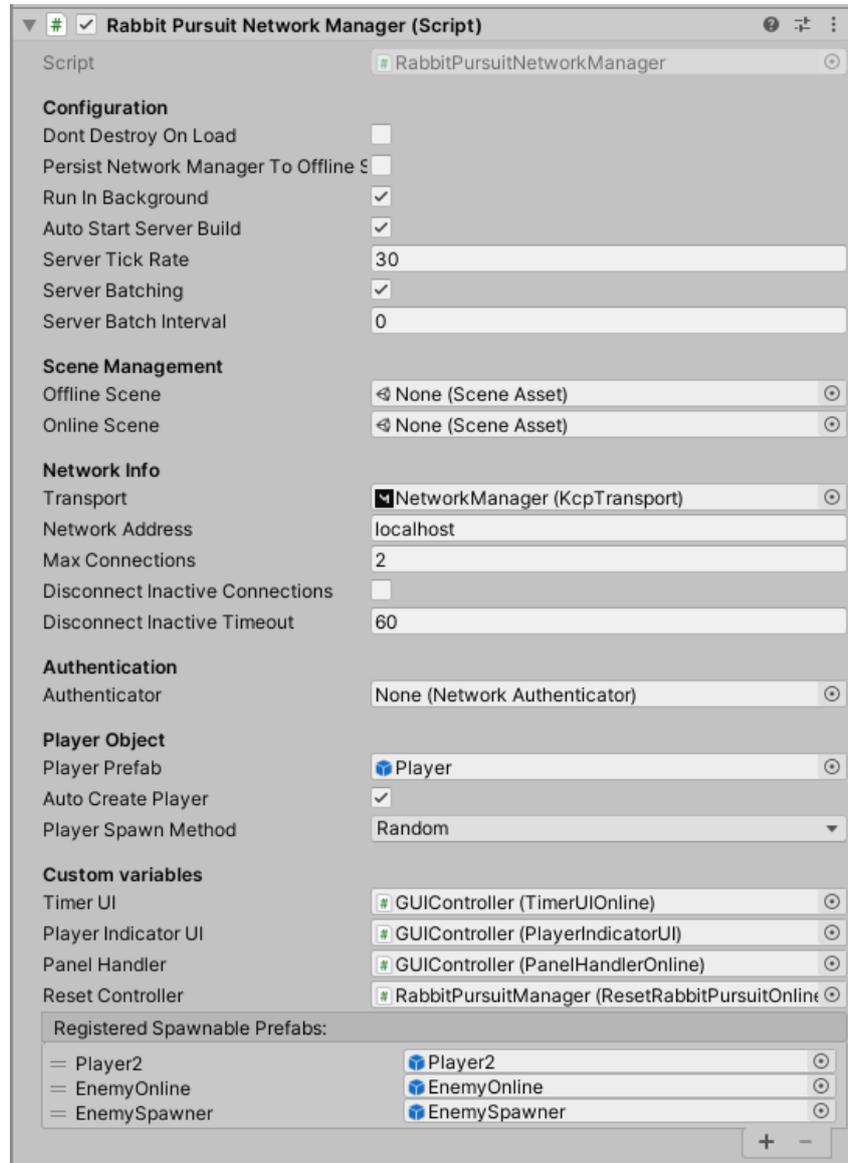


Figura 80. Componente *RabbitPursuitNetworkManager*

Elementos de la red

Diferentes objetos tienen que existir en la red para poder ser identificados y controlados por *Mirror*. Estos objetos necesitan de ciertos componentes proporcionados por la librería para su correcto funcionamiento.

Empezando por los jugadores, estos necesitan al igual que todos los demás objetos un componente *Network Identity* para ser conocidos por *Mirror* y poder funcionar de forma online. Como los jugadores necesitan moverse por todo el mapa necesitan los componentes *Network Transform*, para sincronizar con los demás clientes la posición en la que se encuentran en cada momento como también su escala y rotación, y el componente *Network Rigidbody2D* para sincronizar las

interacciones con el motor de físicas, ya que el jugador se mueve proporcionándole una velocidad e impulso en una dirección. Sumado a todos estos, los personajes poseen animaciones las cuales deben ser sincronizadas también si se quiere homogeneidad entre dispositivos y para ello se ha añadido el componente *Network Animator*. Todos estos componentes se pueden observar en la Figura 81.

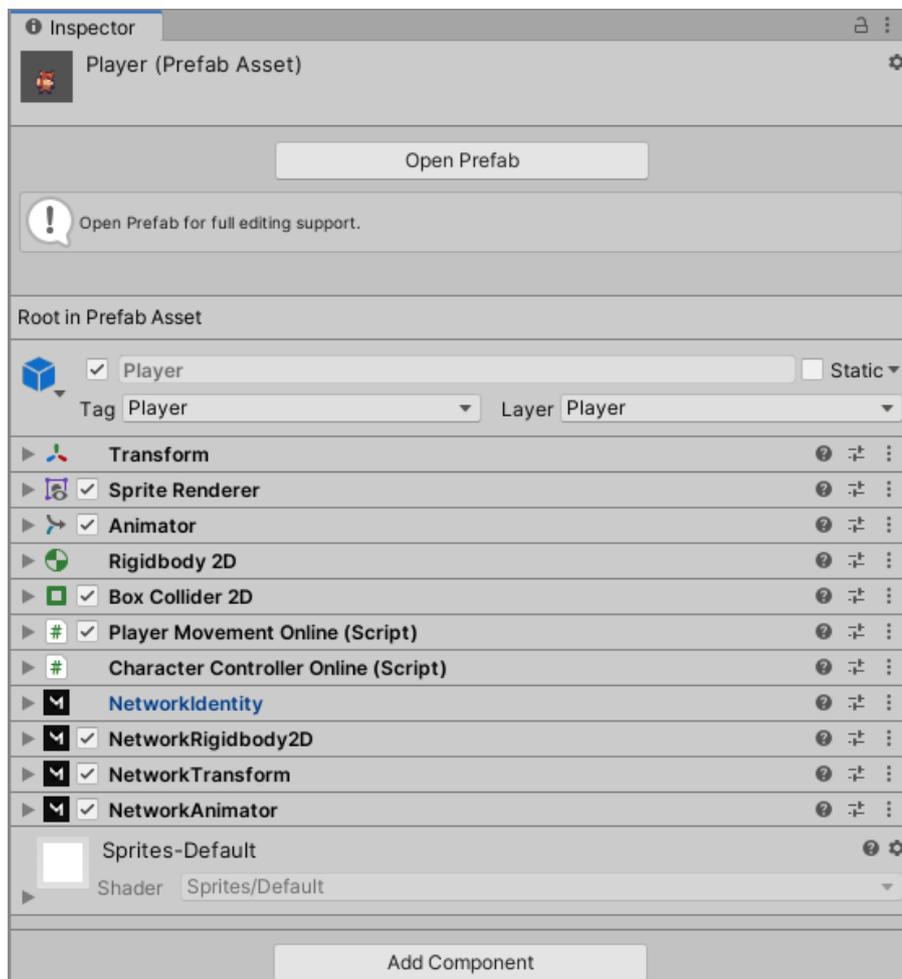


Figura 81. Componentes *Network* del objeto *Player*

El objeto enemigo también se mueve con impulsos y fuerzas por lo que necesita de un *Network Rigidbody2D* para sincronizarse, pero como su movimiento no necesita tanta precisión como el jugador, no se ha incluido un *Network Transform* ni tampoco un *Network Animator* para las animaciones. Como es de esperar, contiene un *Network Identity* para poderse crear a través de la red y tener Mirror control sobre el objeto (Figura 82).

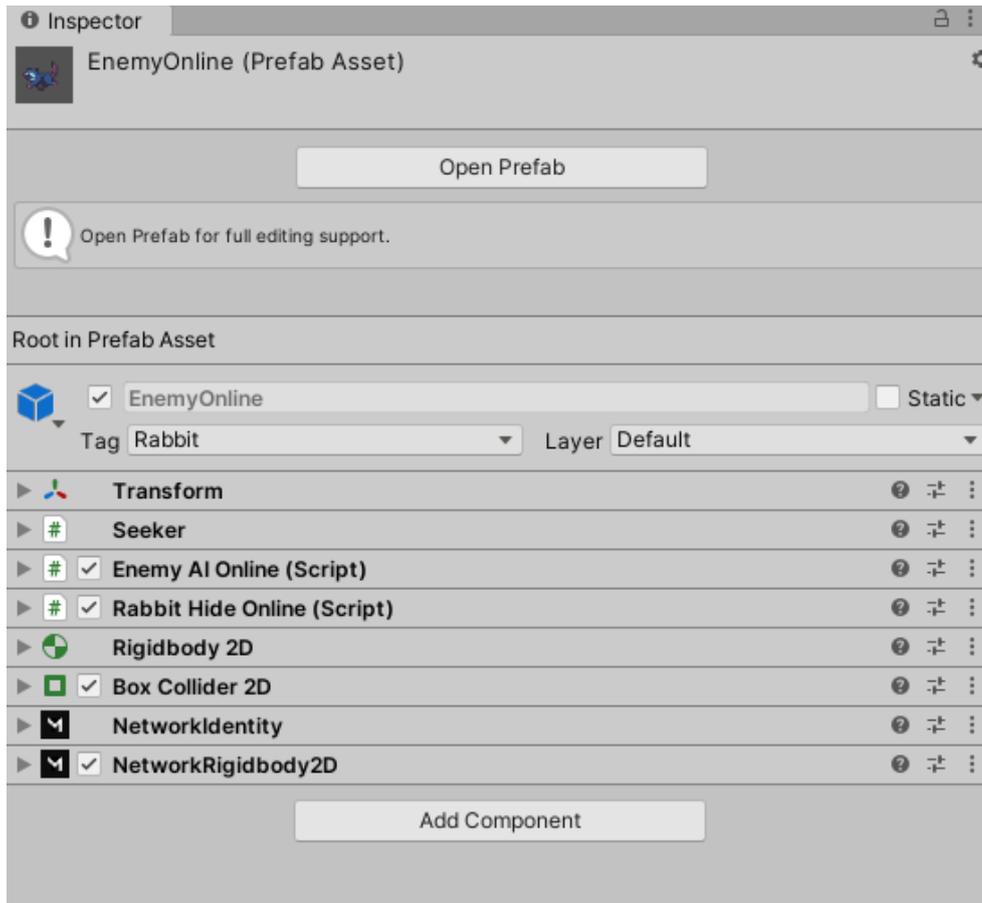


Figura 82. Componentes de red de los objetos enemigos

Por último, tenemos objetos de escena que se encargan de controlar aspectos de la escena y del minijuego, entre los que se encuentran el *ScoreController* (Figura 83) el cual controla las puntuaciones de los jugadores y su propia actualización visual en la interfaz; el *RabbitPursuitManager*, presente en la Figura 84, que necesita sincronizar los *acorns* en el modo online y aspectos del minijuego como el reinicio de posiciones al finalizar la partida; un objeto *GUIController* encargado de gestionar el cronómetro, el panel de victoria o derrota, la interfaz de carga de escena para transitar a la siguiente, así como controlar el indicador para marcar qué personaje maneja el jugador. Todos estos *scripts* que controlan dichas funcionalidades se pueden observar en la Figura 85; y, por último, tenemos un objeto *EnemySpawner* (Figura 86) que se instancia en tiempo de ejecución en el servidor cuando ambos clientes se conectan y están listos, el cual se encarga de la creación de los enemigos que aparecen en el mapa en los distintos puntos de *spawn*. Como se lleva mencionando desde el principio, todos estos objetos precisan de *Network Identity* para poder implementar funcionalidad multijugador.

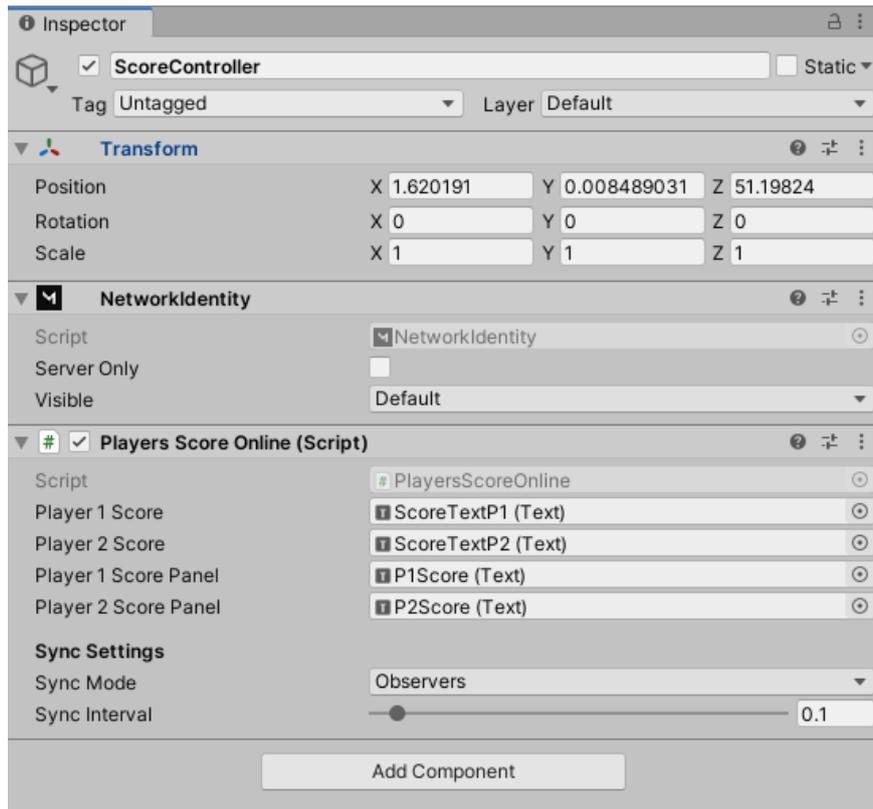


Figura 83. Objeto *ScoreController* y sus componentes

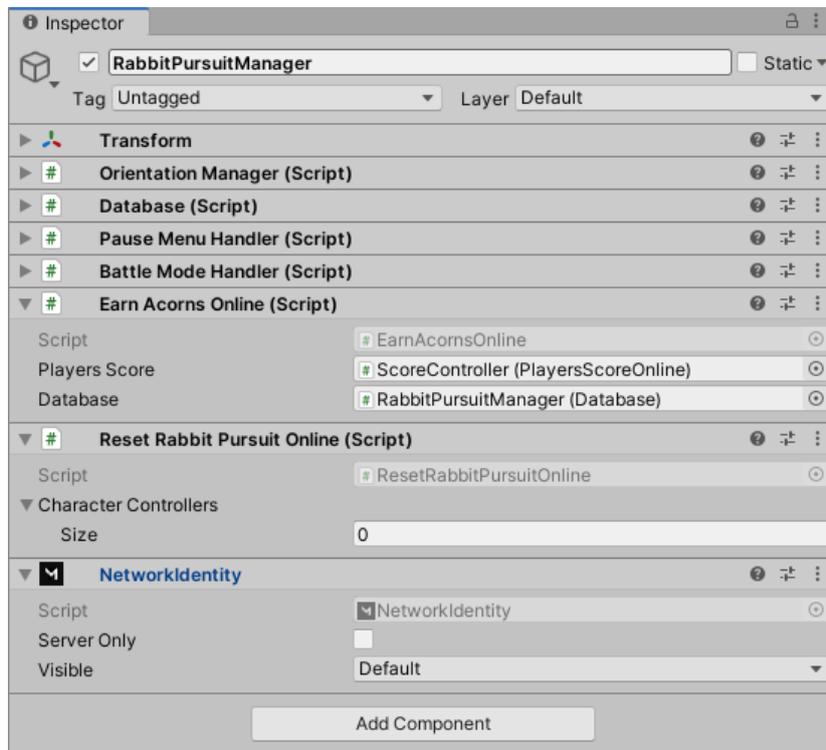


Figura 84. Objeto *RabbitPursuitManager* y sus componentes

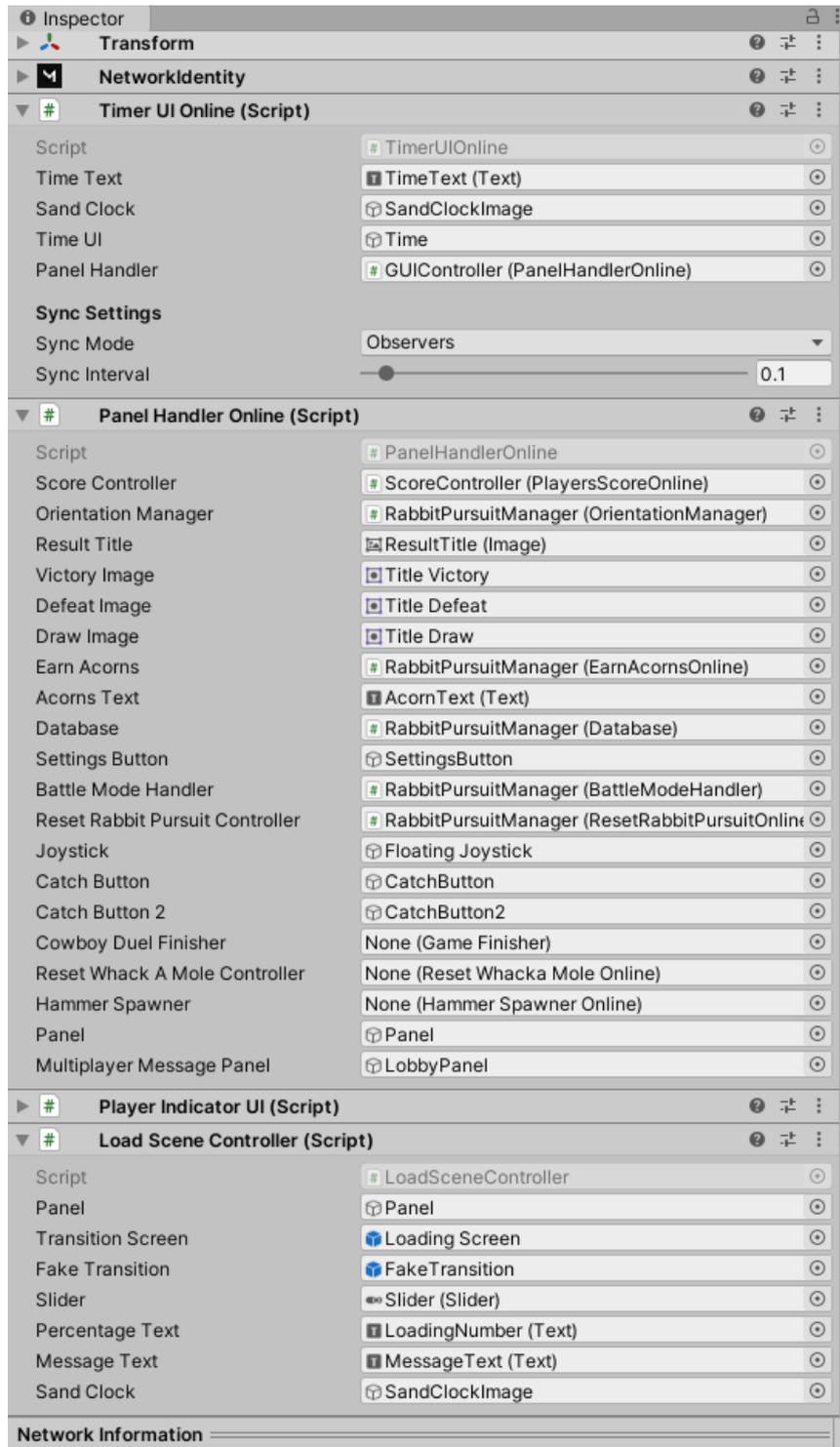


Figura 85. Objeto *GUIController* y sus componentes

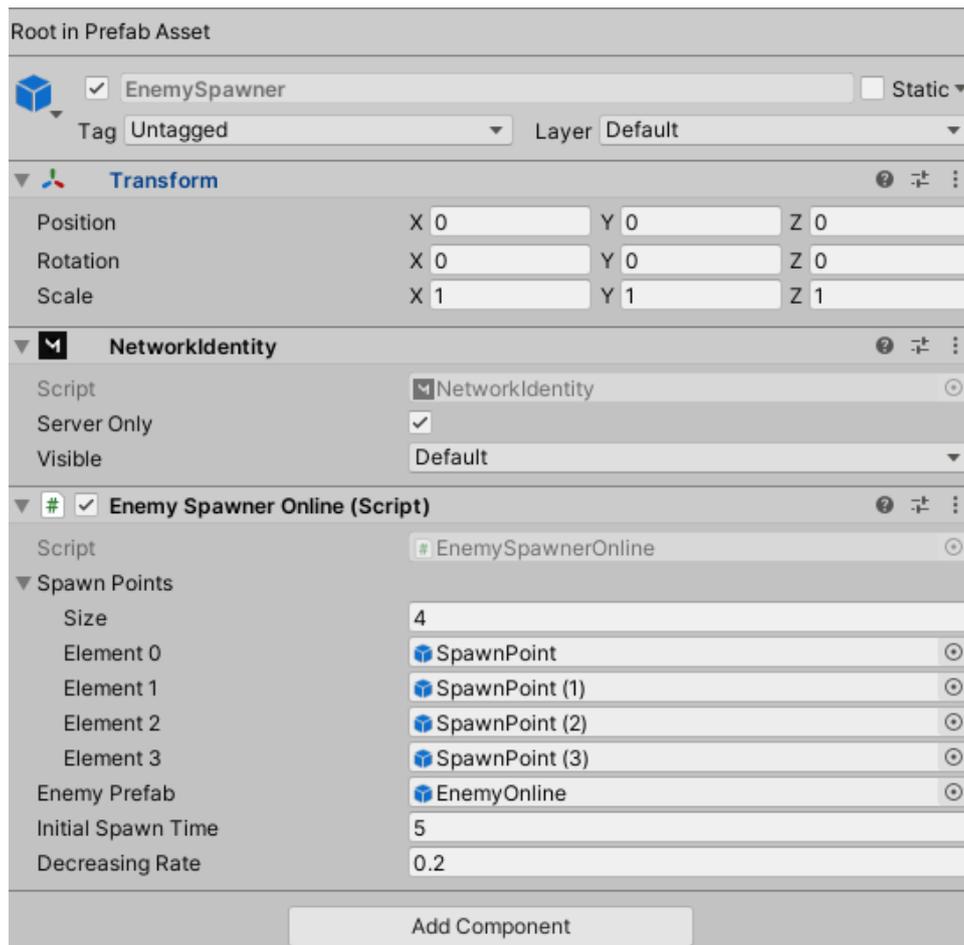


Figura 86. Objeto *EnemySpawner* y sus componentes

7.1.2 Whack-A-Mole

NetworkManager

Este componente (Figura 87) cumple la misma funcionalidad mencionada en el primer apartado del minijuego anterior, pero además se encarga de tareas específicas de *Whack-A-Mole*. Cuando el número de jugadores conectados asciende a dos, el *manager* también deshabilita el panel de búsqueda de usuarios, inicia la animación para indicar a cada jugador qué personaje es el suyo, inicializa y empieza el contador de tiempo, así como habilita los elementos visuales pertenecientes al minijuego. Una vez hecho todo eso, el componente crea los diferentes *spawners* para cada uno de los tres tipos de topes existentes en el nivel. Estos *spawners* se crean a nivel de servidor y no se encuentran activos en los clientes.

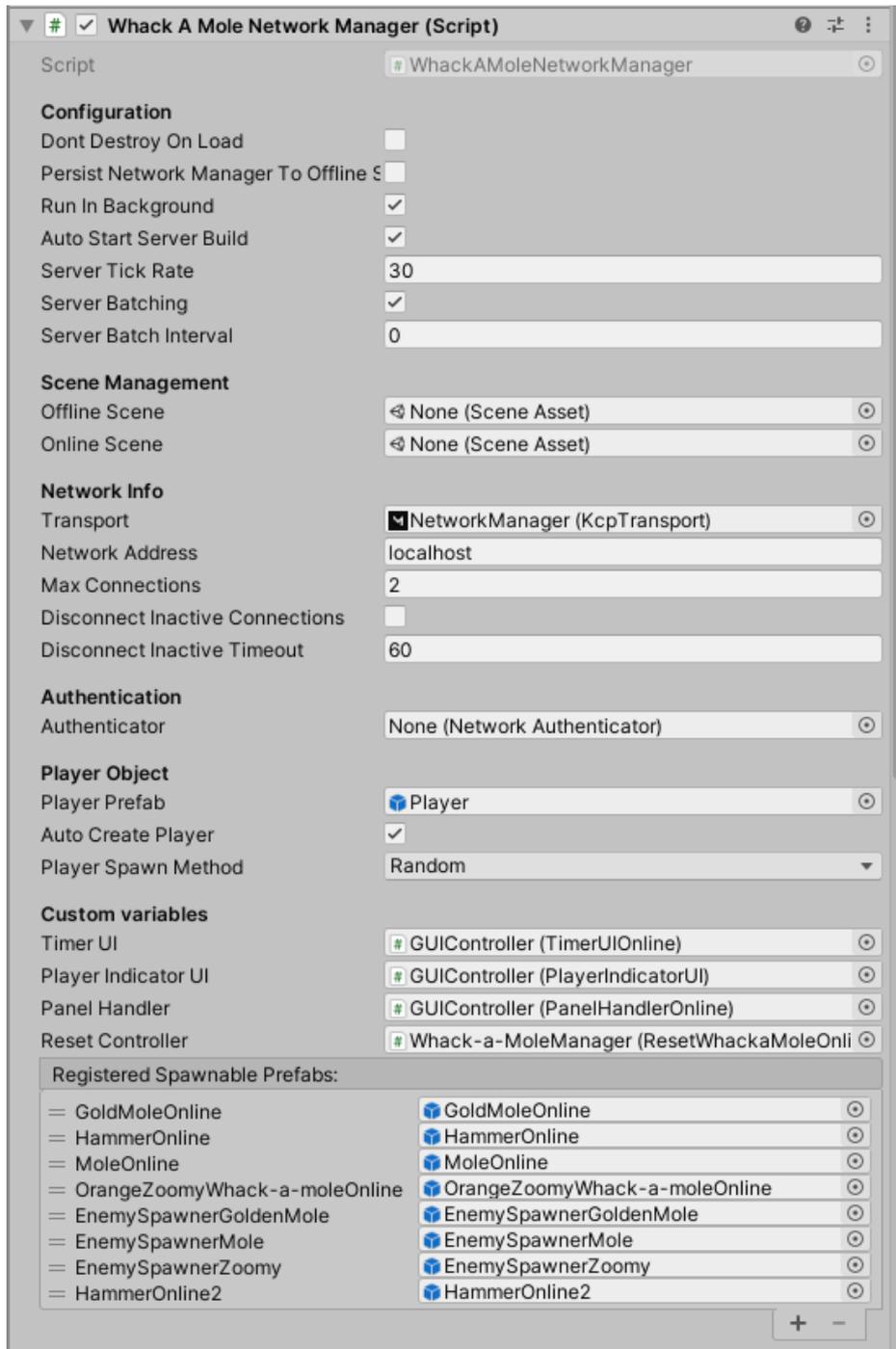


Figura 87. Componente *WhackAMoleNetworkManager*

Elementos de la red

Este minijuego comparte objetos en red con los otros dos minijuegos, como un *manager* para gestionar los *acorns* obtenidos al finalizar la partida llamado *WhackAMoleManager* y un *GUIController* para gestionar el cronómetro, la pantalla final cuando acaba el juego, la pantalla de carga a la siguiente escena o el indicador de personajes de cada jugador.

Cuando se conectan los clientes, el *NetworkManager* hace *spawn* de un objeto jugador. En este caso como el *prefab*, objeto del jugador, necesita ser creado por toda la red, tiene un *NetworkIdentity*, además de su script para realizar la funcionalidad de crear mazos en los lugares pulsados (Figura 88). Ambos clientes tienen identificado su objeto jugador asociado para diferenciar entre ambos mazos al golpear.

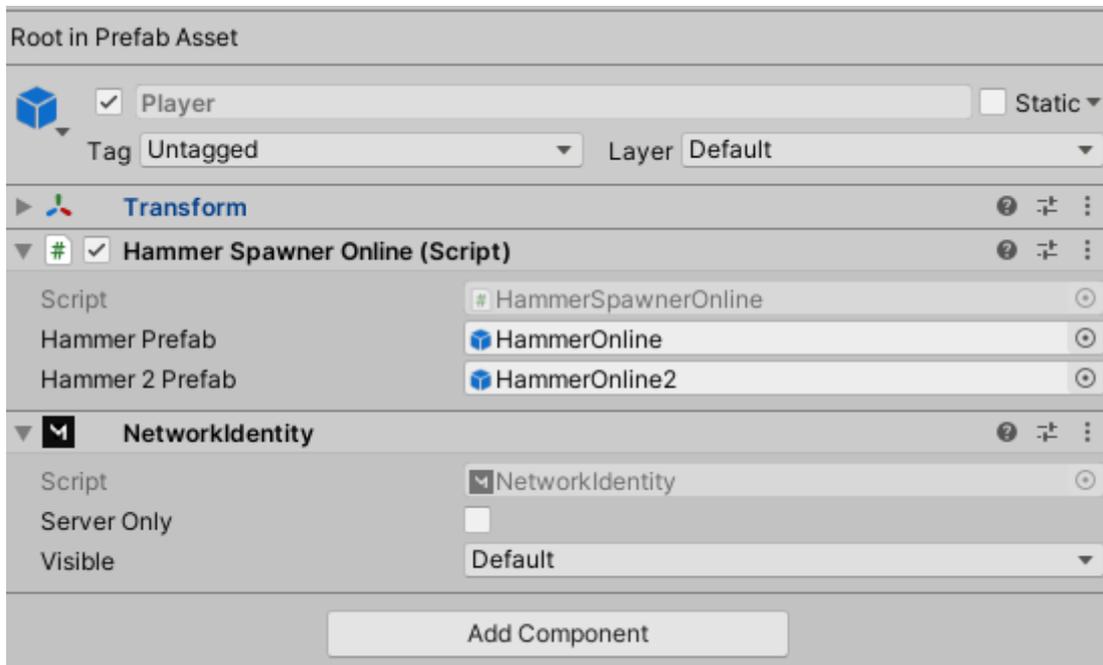


Figura 88. *Prefab* del jugador con sus componentes

Los clientes, cuando tocan en algún lugar de la pantalla, instancian un mazo, el cual debe tener a su vez un *NetworkIdentity* para existir en la red. Como este objeto colisiona con el motor de físicas de *Unity*, necesita un *Network Rigidbody2D* para que dichas colisiones estén sincronizadas en todas las instancias de clientes como se observa en la Figura 89.

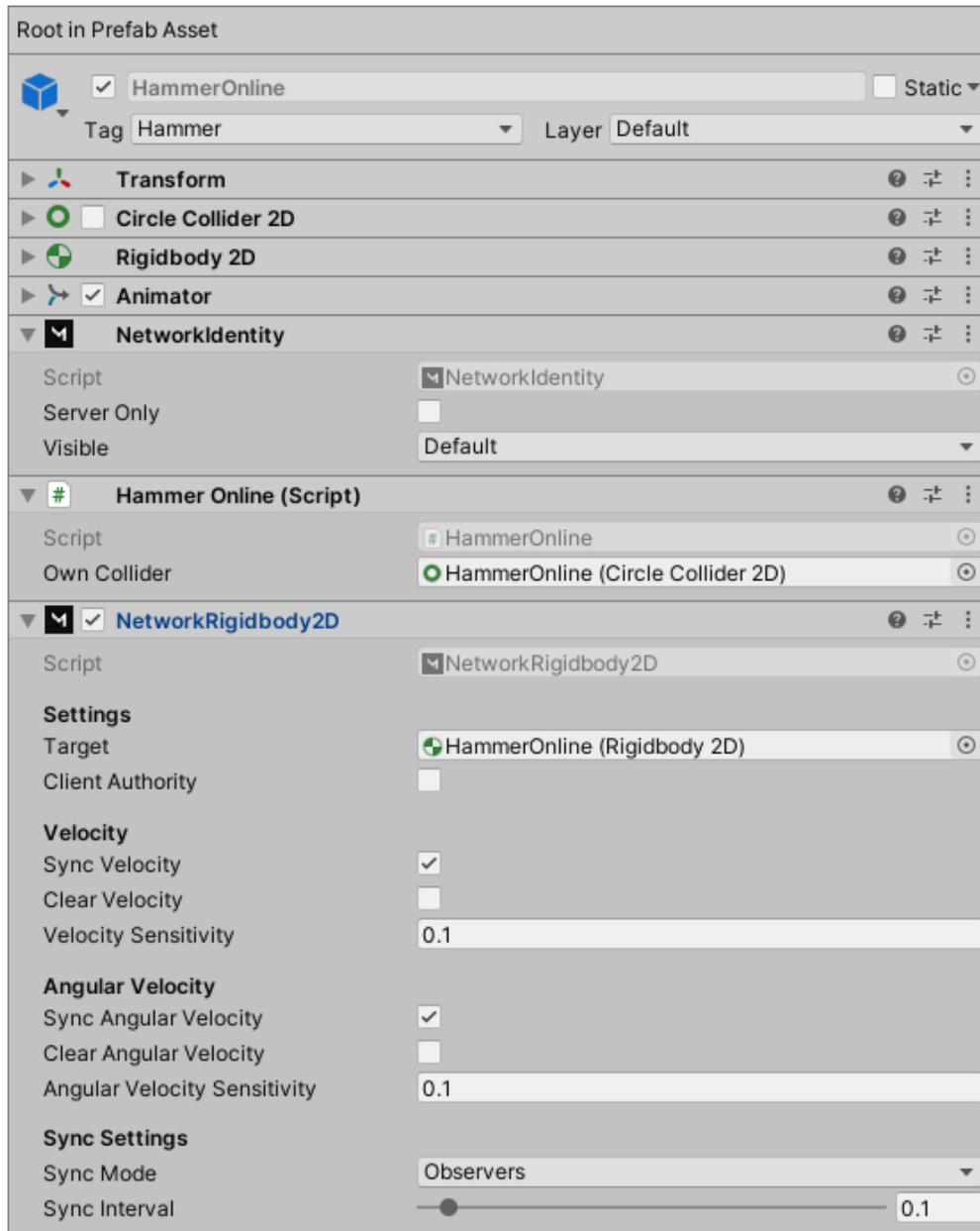


Figura 89. Prefab del mazo con sus componentes

Existen tres tipos de enemigos en este minijuego, pero todos comparten los mismos componentes y solo se distinguen por sus *sprites*. Simplemente precisan del *NetworkIdentity* (Figura 90) para funcionar dentro de la red de *Mirror*.

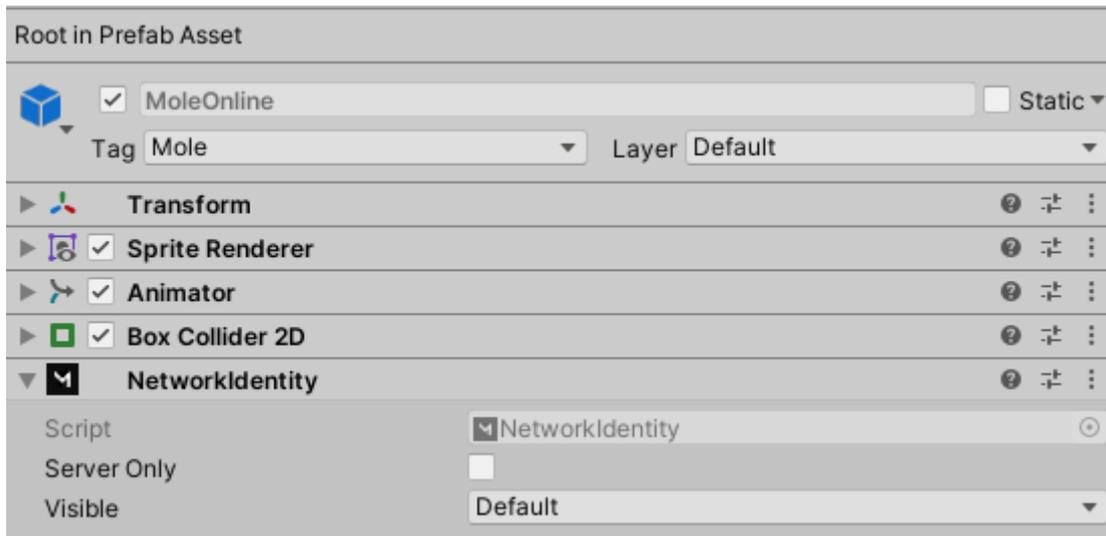


Figura 90. Prefab de tipo de enemigo topo estándar

Por último, el minijuego tiene tres tipos de *spawners*, uno para cada variante de enemigo. Estos objetos contienen necesariamente, como se ha mencionado anteriormente, un *NetworkIdentity* y tienen marcada la casilla de *Server Only*, mostrado en la Figura 91, porque nos interesa que solo existan en el lado del servidor, y este sea el encargado de crear los diferentes enemigos e instanciarlos a través de la red. Además, este objeto junto a los otros se crea en tiempo de ejecución y obtiene referencias de los distintos puntos de creación disponibles.

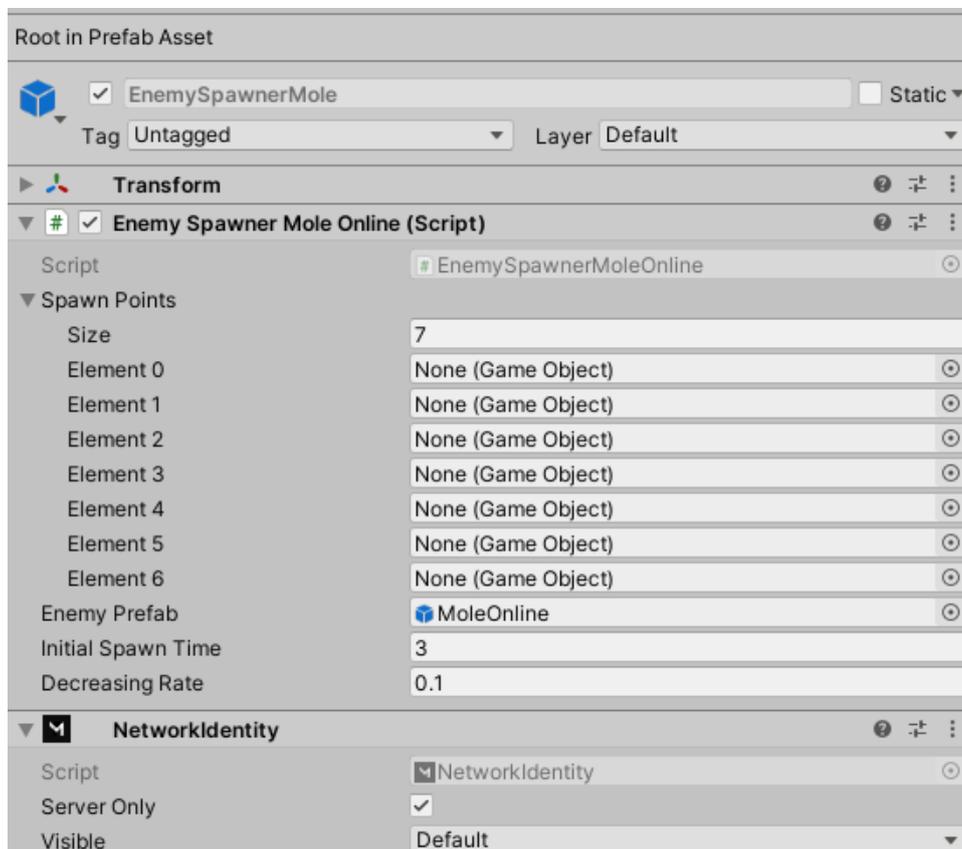


Figura 91. Prefab del *spawner* de enemigos de tipo topo comunes

7.1.3 Cowboy Duel

NetworkManager

El *NetworkManager* de *CowboyDuel* (Figura 92) además, habilita los componentes *Animator* en cada uno de los objetos jugador. Esta acción es necesaria para que, el minijuego basado en animaciones tenga cada una de ellas sincronizadas y se puedan llevar a cabo cada uno de los eventos del nivel. También se encarga de indicar a cada jugador su personaje jugable además de mostrar elementos de la interfaz propia de este minijuego.

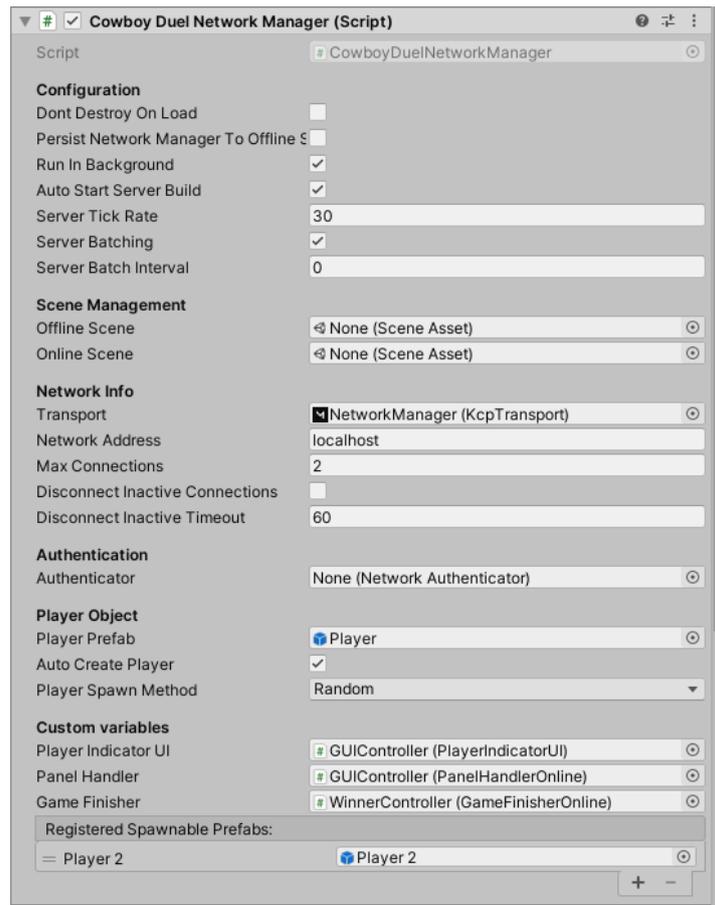


Figura 92. Componente *CowboyDuelNetworkManager*

Elementos de la red

Los jugadores cuentan con su *prefab* propio (Figura 93), cada uno de ellos con un *Sprite* visual distinto, pero compartiendo toda la funcionalidad y componentes de red. Como todos los objetos que representan los jugadores en la red, tienen adjuntos un componente *NetworkIdentity* para funcionar dentro del multijugador. El hecho de que el minijuego se base en una sucesión de animaciones que pueden accionar eventos, hace que no se precise de ningún objeto más específico. Comparte con los minijuegos anteriores los ya mencionados *GUIController* y el *manager* del nivel llamado *CowboyDuelManager*.

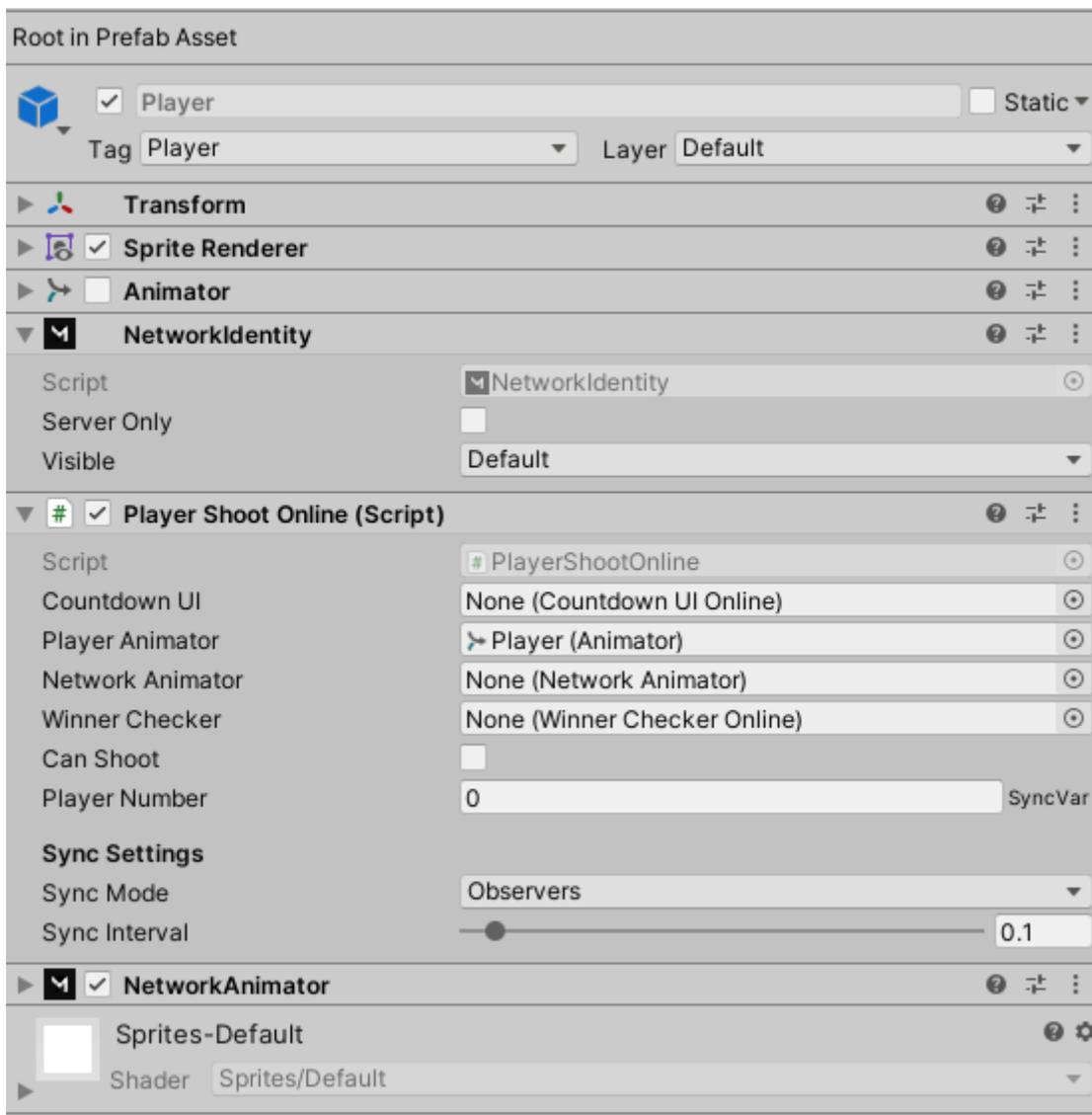


Figura 93. Prefab jugador *CowboyDuel* y sus componentes

8. Resultados

8.1 Encuestas

Al finalizar el MVP2, se subió el videojuego a la *Play Store* de *Google* para que los usuarios pudieran probarlo y darnos *feedback*. Además, se proporcionó una encuesta para que pudieran opinar sobre el juego y qué les había parecido esta nueva funcionalidad multijugador. Esto sirvió para recoger problemas, mejoras que nos recomendaban incluir para ampliar la experiencia de juego o simplemente si les había gustado las nuevas características. Entre la encuesta encontramos dos preguntas relacionadas con el multijugador de esta entrega.

La primera, fue una pregunta que se puede representar visualmente (Figura 94) y trata sobre si les ha gustado la implementación del sistema multijugador en la última actualización y su mecánica de duelo 1vs1 contra otra persona. Podemos destacar la satisfacción general de los usuarios por la inclusión de esta mejora respecto a un solo jugador, donde 18 de los 21 encuestados, es decir, el 85,6% del total, puntuó por encima del ocho en una escala sobre diez. Al resto de encuestados le resultó indiferente esta inclusión. Con esta pregunta concluimos que ha sido muy buena idea el añadido de la faceta multijugador al juego.

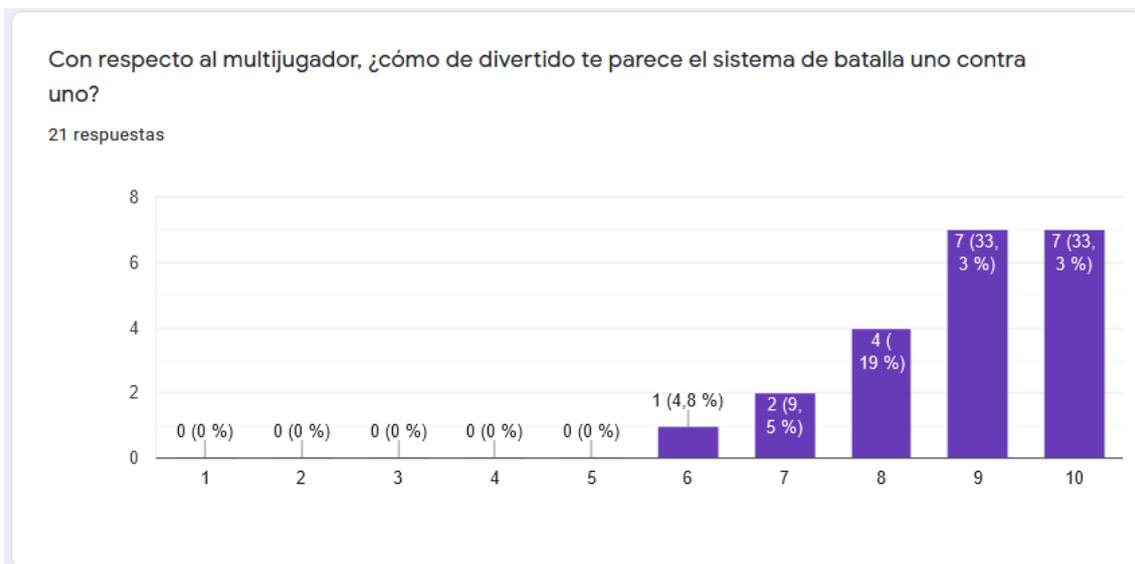


Figura 94. Satisfacción sistema de batalla 1vs1 del multijugador

Por otra parte, se le preguntó a los familiares, amigos y conocidos para saber si tenían alguna sugerencia que hacer sobre el multijugador y así poder ir mejorando continuamente la experiencia del usuario. Entre las respuestas, encontramos una posible mejora y es la de incluir en un futuro la posibilidad de poder jugar con personas de alrededor del mundo y no solo entre amigos y familiares que se encuentren en el mismo WIFI. Esto podría ser una mejora sustancial y que

cambiaría drásticamente la experiencia de juego para ofrecer un servicio global y competitivo que puede animar y avivar el juego.

Además, se sugirió una mejora en el sistema de emparejamiento de jugadores, por la cual pasaría a integrar un sistema de *matchmaking* más complejo que emparejaría dos jugadores y cuando estuvieran listos seleccionaría de forma aleatoria a uno de los dos para hacer de *host* mientras que el otro simplemente se conectaría. Esto actualizaría y potenciaría el funcionamiento actual, que se rige por un *matchmaking* sencillo, donde un jugador busca si existe una partida y si no la crea. Por el momento, esto ocasiona el problema de que un jugador debe iniciar la búsqueda con tiempo para poder crearla y el otro cliente pueda encontrarla y unirse.

8.2 Pruebas de Aceptación

Antes de lanzar la app a producción y publicarla en la *Play Store* de *Google*, se realizaron pruebas de aceptación (PA) para ver si se cumplían los requisitos y las funcionalidades esperadas al inicio de la planificación del multijugador.

Por cada minijuego se realizaron PAs, todas ellas pasadas y validadas por otros miembros del equipo distintos al desarrollador de dicha funcionalidad. Entre las varias PAs, podemos encontrar el ejemplo de la PAs validada para el minijuego de *Binky Pursuit* una vez finalizada su funcionalidad multijugador (Figura 95). La primera prueba era validar que los objetos pertenecientes a la escena del minijuego estuvieran sincronizados y hubiera consistencia entre los elementos, mientras que la segunda, se pedía probar el correcto funcionamiento de la puntuación, donde a cada jugador se le sumaba correctamente un punto al capturar un *binky* y no hubiera errores de sumar los puntos incorrectos o sumarle al otro jugador su puntuación.

	5835 - Sincronización correcta del estado del minijuego online	26/5/2021 18:17	2773	Sprint 4			8/6/2021 13:06
	5831 - Sincronización de puntuación online	26/5/2021 18:10	2773	Sprint 4			8/6/2021 12:25

Figura 95. Pruebas de aceptación del multijugador de *Binky Pursuit*

Respecto al segundo minijuego *Whack-A-Mole* (Figura 96), encontramos una PA dedicada al correcto funcionamiento del sistema de golpear topos en el que, ambos jugadores, al encontrarse en la misma partida, deben crear cada uno su propio mazo diferenciado y ser capaces de golpear algún topo y este realizar sus correspondientes animaciones. Todo esto debe estar sincronizado a través de la red. La segunda PA va dedicada al cálculo correcto de la puntuación al golpear un topo, sumando diferentes puntos dependiendo de si ha golpeado al topo normal o al dorado, o si

por el contrario ha golpeado un *Zoomy* lo cual le restaría puntos. Estos puntos deben sumarse o restarse al jugador que ha golpeado y esta PA sirve para validar dicho requisito.

	5842 - Sincronización correcta del estado de los personajes	26/5/2021 18:32	2774	Sprint 4			29/6/2021 11:24
	5836 - Sincronización de puntuación online	26/5/2021 18:22	2774	Sprint 4			29/6/2021 11:24

Figura 96. Pruebas de aceptación de *Whack-A-Mole*

En cuanto al tercer minijuego *Cowboy Duel* y estar este basado en animaciones, se realizó una PA (como se puede observar en la Figura 97) para validar el correcto funcionamiento de las animaciones de ambos personajes, especialmente la de disparo. Este hecho es de vital importancia probarlo ya que, una pequeña desincronización con las animaciones de uno de los personajes podía desencadenar un mal funcionamiento del minijuego y privar a este de jugabilidad alguna, por muy minúsculo que fuera el margen de error. Es por eso por lo que se probó exhaustivamente que cada una de las animaciones de disparo estuvieran bien cronometradas y sincronizadas entre dispositivos y asegurar el funcionamiento de *Cowboy Duel*, ya que de esta sincronización dependía el cálculo de tiempo sobre que jugador había disparado antes y a quien se elegía ganador de la ronda.

	5839 - Comparación de los tiempos de disparo correcta	26/5/2021 18:26	2775	Sprint 4			15/6/2021 17:56
---	---	-----------------	------	----------	---	---	-----------------

Figura 97. Prueba de aceptación de *Cowboy Duel*

9. Conclusiones

Si nos paramos a observar los objetivos establecidos en el apartado 1.3, se han cumplido dichos objetivos marcados al inicio del proyecto. El equipo planteó una serie de metas que se debían realizar y se han llevado a cabo dentro del margen de tiempo, pero sin omitir una serie de problemas que han surgido como puede pasar en cualquier equipo de desarrollo.

El equipo necesitaba una persona que se dedicara al arte, y para ello se realizó una búsqueda en internet para finalmente encontrar un quinto integrante al proyecto procedente del extranjero. Al principio, el desarrollo del juego seguía su curso pero dicha persona, al no estar sujeta al proyecto, y habiendo acordado su ayuda siempre que quisiera y pudiera, se desatendió y no pudo participar más en el apartado visual del proyecto. Todo esto ocasionó que el integrante del equipo encargado de desarrollar la parte del *front-end* tuviera que realizar unos diseños para el arte dentro de sus posibilidades con ayuda del resto de los miembros. A este hándicap se han sumado ciertos problemas técnicos relacionados con el proyecto, donde han sido notorios en el apartado multijugador, debido a la inexperiencia en dicho campo.

Sim embargo, se ha conseguido desarrollar dos MVPs funcionales que cumplen las expectativas. Dichos MVPs han sido sometidos a diferentes experimentos y los usuarios han podido proporcionar un feedback clave, para ir mejorando poco a poco el producto final.

Como los cuatro integrantes provenimos de la rama de Ingeniería del *Software*, se ha organizado el proyecto con una facilidad mayor a la que pueden tener otro tipo de proyectos, siguiendo metodologías ágiles y estructurando el juego en diferentes tareas a ser repartidas entre los miembros.

Si se mira atrás cuando se decidió hacer de este proyecto una idea de emprendimiento, se observa la satisfacción del equipo de haber sido capaz de desarrollar el juego dentro del marco y poder lanzarlo en la *Google Play Store* de *Android*, estando disponible en ni más ni menos que 53 países. Esta experiencia beneficia a los integrantes para futuros proyectos que puedan realizar.

Por último, en cuanto al apartado multijugador, se ha podido alcanzar la meta de desarrollar un juego multijugador el cual permite la interacción de dos personas en los minijuegos ofreciendo una faceta competitiva inexistente en su versión contra la IA. Es verdad que han surgido problemas a la hora de realizarlo como se ha mencionado en el primer párrafo, pero el equipo no se desanima y sigue su desarrollo para brindar una experiencia multijugador con jugadores que se encuentren en diferentes partes del mundo.



10. 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 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.

11. Apéndices

11.1 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).

RPC: Remote Procedure Call (Llamada de Procedimiento Remotas, en español).

11.2 Lista de recursos gráficos

Diseñados por los miembros del equipo de este TFG:

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



Obtenidos de otras fuentes:

- Zorros y sus animaciones en *Binky Pursuit* y *Cowboy Duel*:
<https://assetstore.unity.com/packages/2d/characters/sunny-land-103349>
- *Binkies* y sus animaciones en *Binky Pursuit*:
<https://assetstore.unity.com/packages/2d/characters/sunny-land-103349>
- Red de *Binky Pursuit*: https://favpng.com/png_view/net-the-legend-of-zelda-skyward-sword-animal-crossing-new-leaf-the-legend-of-zelda-ocarina-of-time-the-legend-of-zelda-a-link-to-the-past-png/KM8beq3G
- 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>

11.3 Game Design Document (GDD)



Game Design Document para:



All work Copyright ©2021 by SwifterGames

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

Version #3.0

Domingo, 20 de Junio de 2021

1. Historial de diseño.....	100
Versión 1.00	100
Versión 2.00	100
Versión 3.00	100
2. Visión global del juego	100
Filosofía.....	100
Preguntas frecuentes.....	100
3. Conjunto de características.....	101
Características generales	101
Características multijugador.....	101
Editor.....	102
Jugabilidad	102
4. El mapa del mundo.....	102
Visión general	102
Zonas temáticas	102
Parque en constante evolución	103
Sistema de renderizado.....	103
Cámara	103
Motor de videojuegos.....	104
5. Personajes del juego	104
Visión general	104
Creando un personaje	105
Personajes especiales para cada minijuego	105
6. Interfaz de usuario.....	105
Visión general	105
7. Herramientas	108
Visión general	108
Variedad de herramientas.....	108
8. Banda sonora y efectos de sonido	109

Visión general	109
Sonido 3D.....	109
9. Minijuegos.....	109
<i>Binky Pursuit</i>	109
<i>Whack-A-Mole</i>	112
<i>Cowboy Duel</i>	115
10. Juego para un jugador	117
Visión general	117
Historia.....	117
11. Juego multijugador.....	118
Visión general	118
Batalla de minijuegos.....	118
Máximo de jugadores.....	118
Servidores.....	118
Personalización.....	118
Internet	118
Persistencia.....	119
Guardado y cargado	119
12. Calendario de trabajo	119



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



Figura 98. 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 99 se observa el diseño inicial del menú principal del videojuego.



Figura 99. 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 100). 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.

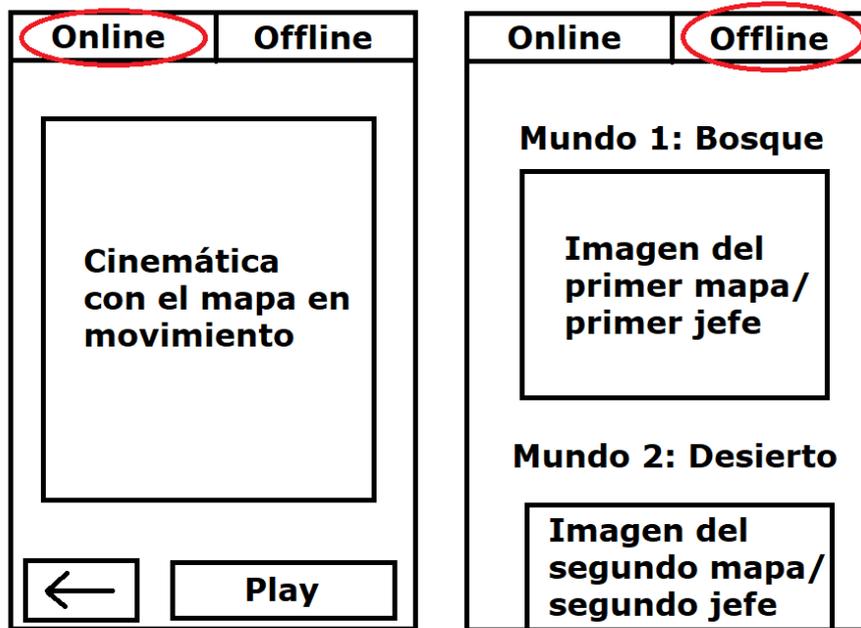


Figura 100. Pantallas de juego

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



Figura 101. 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 102) 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.

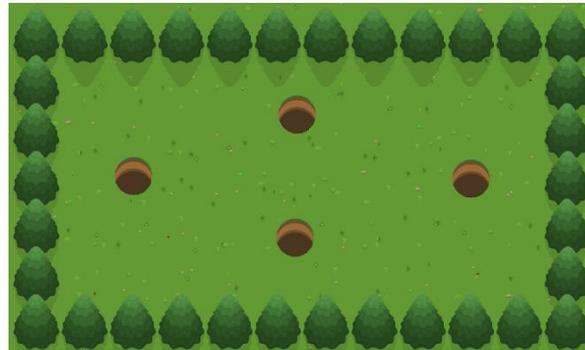


Figura 102. Escenario de *Binky Pursuit*

Interfaz de usuario

Como se aprecia en la Figura 103, 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.

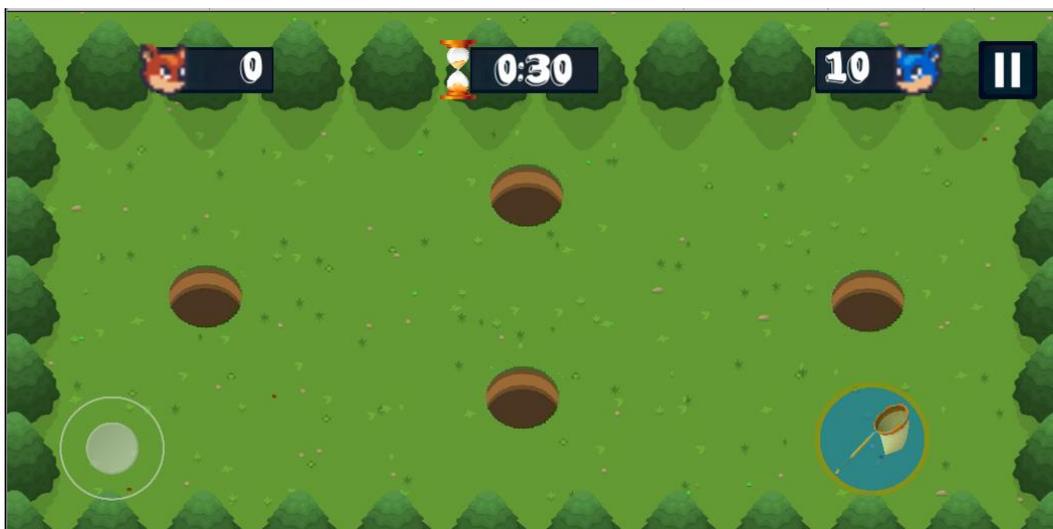


Figura 103. Interfaz de *Binky Pursuit*.

Personajes

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

- Zorros (Figura 104): Los personajes controlables por los jugadores.



Figura 104. Zorros de *Binky Pursuit*

- *Binkies* (Figura 105): 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 105. *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 106) simula una llanura donde hay siete madrigueras de topos. En los bordes del mapa existen vallas que delimitan el mapa.

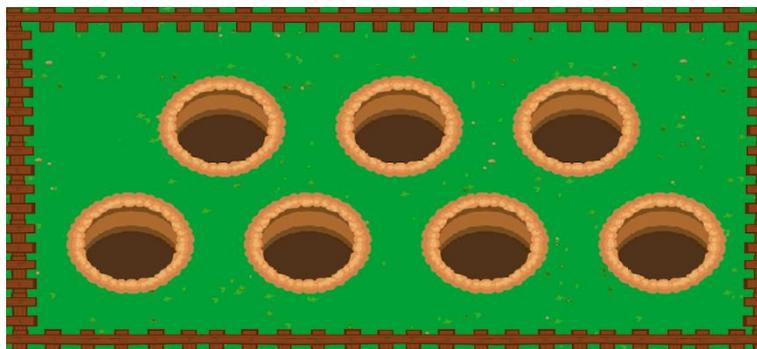


Figura 106. Escenario de *Whack-A-Mole*

Interfaz de usuario

Como se puede ver en la Figura 107, 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 107. 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 108): es el personaje que controla el jugador. Simula el martillo real del arcade original *Whack-A-Mole* (Guacamole en España).



Figura 108. Martillo de *Whack-A-Mole*

- Topo (Figura 109): es el enemigo estándar. Golpearle con el martillo otorga un punto.



Figura 109. Topo normal de *Whack-A-Mole*

- Topo dorado (Figura 110): es un topo especial que tiene menor probabilidad de aparecer que el topo. Golpearle con el martillo otorga cinco puntos.



Figura 110. Topo dorado de *Whack-A-Mole*

- *Zoomy* (Figura 111): es el personaje del jugador. Golpearle con el martillo resta tres puntos.



Figura 111. *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 112) simula un desierto del lejano oeste donde los vaqueros solían tener este tipo de duelos.



Figura 112. Escenario de *Cowboy Duel*

Interfaz de usuario

Como se puede apreciar en la Figura 113, 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 113. Interfaz de *Cowboy Duel*

Personajes

En este minijuego participan los dos zorros, uno de cada jugador (Figura 114).



Figura 114. 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 115. Calendario de trabajo

12. 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>>

