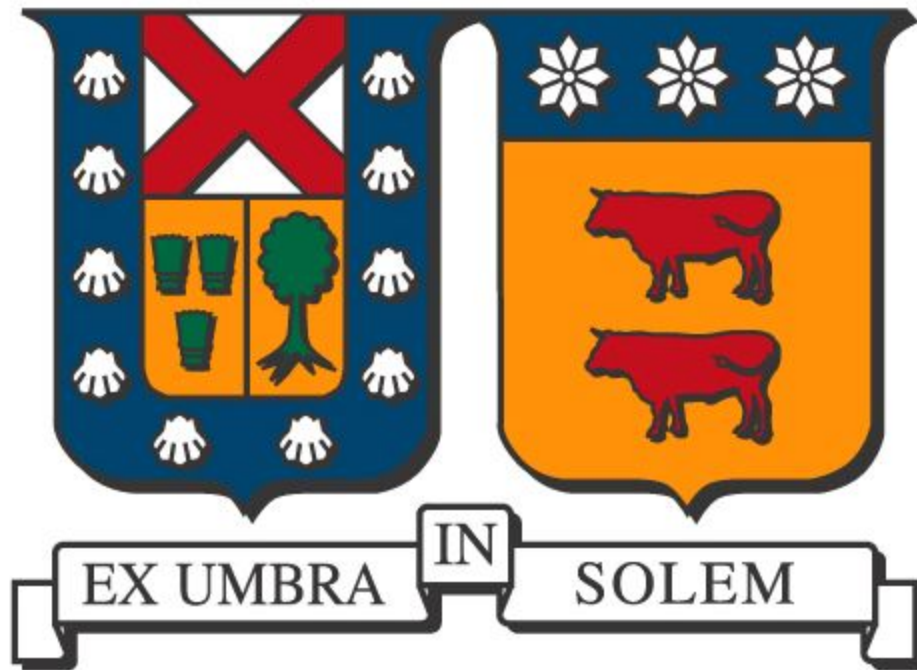


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



DISEÑO Y DESARROLLO DE VIDEOJUEGO HIPER-CASUAL EN
UNITY PARA LA PLATAFORMA ANDROID

MATÍAS HERNÁN YAÑEZ POHL

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO DE
EJECUCIÓN EN INFORMÁTICA

Profesor guía: CECILIA REYES COVARRUBIAS
Profesor correferente: SVEN VON BRAND LAREDO

FECHA 2019

Dedicatoria

Agradezco en un bucle infinito a mi familia por la paciencia, comprensión y amor que me han entregado a lo largo de mi vida. Agradezco a mis amigos y compañeros que compartieron esta etapa conmigo y llenaron de buenas historias mi vida universitaria. Y por último agradecer a Mane por creer en mi capacidad y porque siempre me has entregado comprensión, cariño y amor. Le amo.

Resumen

Dentro del amplio abanico de juegos móviles, un género se ha apoderado de los rankings de descargas con una alta tasa de crecimiento: los juegos hiper-casuales. Estudios demográficos y de comportamiento de los jugadores explican por qué estos juegos de mecánicas sencillas y jugabilidad instantánea, han sido un éxito estos últimos años. Este trabajo busca diseñar y desarrollar un videojuego hiper-casual para dispositivos móviles usando el software Unity, con un equipo multidisciplinario y utilizando una metodología ágil.

Palabras Clave: Videojuegos; Hyper-Casual; Unity3D; Android

Abstract

Within the wide range of mobile games, a genre has taken over the rankings of downloads with a high growth rate: hyper-casual games. Demographic and player behavior studies explain why these games of simple mechanics and instant gameplay, have been a success in recent years. This work seeks to design and develop a hyper-casual video game for mobile devices using the Unity software, with a multidisciplinary team implementing an agile methodology.

Keywords: Video games; Hyper-Casual; Unity3D; Android

Índice de Contenidos

Dedicatoria	1
Resumen	2
Abstract	2
Índice de Contenidos	3
Índice de figuras	6
Índice de Tablas	9
Introducción	10
Capítulo 1: Definición de la oportunidad	11
1.1 Historia de los videojuegos móviles	11
1.2 ¿Qué son los juegos hiper-casuales?	14
1.2.1 Principales mecánicas	17
1.3 Mercado de los juegos móviles	20
1.3.1 Auge de los juegos Hiper-Casuales en el mercado	22
1.3.2 Caracterización del usuario que descarga juegos hiper-casuales	24
1.3.3 Industria de videojuegos en Chile	25
1.4 Equipo de Trabajo	26
1.5 Objetivos de la memoria	27
1.5.1 Objetivo Principal	27
1.5.2 Objetivos Específicos	27
Capítulo 2: Etapa de pre-producción de un videojuego	28
2.1 Introducción	28
2.1.1 Concepto del juego	30
2.1.2 Características principales	30
2.1.3 Género	30
2.1.4 Público dirigido, propósito y requerimientos	30
2.1.5 Jugabilidad	31
2.1.6 Alcance	31
2.2 Especificaciones	32
2.2.1 Plataforma	32
2.2.2 Requerimientos técnicos	32
2.3 Mecánicas y componentes de juego	33

2.3.1 Jugador	33
2.3.2 Escenario	34
2.3.3 Personaje	35
2.3.4 Control del jugador	35
2.3.5 Movimientos del escenario	36
2.3.6 Interacción entre elementos	36
2.4 Interfaz	37
2.4.1 Flujo de juego	37
2.4.1.1 Menú Principal	38
2.4.1.2 Ventana de cuenta regresiva	39
2.4.1.3 Nivel	40
2.4.1.4 Splash Screen	41
2.4.1.5 Video de introducción	41
2.4.1.6 Ventana de derrota	41
2.5 Arte	42
2.5.1 Estilo	42
2.5.2 Arte conceptual	42
2.5.3 Sonido y música	45
2.5.4 Animaciones	45
2.5.5 Fuentes	46
2.5.6 Sistema de partículas y efectos especiales	46
2.6 Calendarización	46
Capítulo 3: Etapa de producción de videojuego	48
3.1 Metodología de desarrollo	48
3.1.1. Kanban y su caracterización en el equipo de trabajo	48
3.1.3 Software de planificación	50
3.2 Producción	50
3.2.1 Núcleo del juego	52
3.2.2 Flujo de juego	54
3.2.3 Publicidad y corrección de errores	55
3.2.4 Sonido, video y efectos especiales.	57
3.2.5 Problemas y optimización	58
3.2.6 Testing	60
Capítulo 4: Etapa de publicación de videojuego	61
4.1 Android	61
4.2 Herramientas de analítica	63

4.2.1 Google Play Console	63
4.2.2 Panel de control de Unity	64
4.2.3 Retroalimentación de usuarios	64
4.2.3.1 Reseñas	64
4.2.3.2 Encuesta	67
4.2.4 Versiones posteriores	68
Capítulo 5: Conclusiones	69
5.1 Mercado	69
5.2 Pre-producción	73
5.3 Producción	75
5.4 Publicación	77
5.5 Proyecciones y trabajo futuro	79
5.6 Reflexiones personales	80
Referencias	83

Índice de figuras

Capítulo 1: Definición de la oportunidad

Figura 1.0: Tower Bloxx, juego hiper-casual del año 2005.	11
Figura 1.1: 100 juegos con mayor recaudación en EEUU.	12
Figura 1.2: Gasto por jugador en el Top 25 juegos móviles el año 2015.	13
Figura 1.3: Frogger (1981) y Crossy Road (2014).	14
Figura 1.4: Menú principal Helix Jump, jugabilidad y sistema de anuncios.	15
Figura 1.5: Tiro perfecto en Baseball Boy.	17
Figura 1.6: Pantalla de juego de 2048.	18
Figura 1.7: Tiempos de espera en Idle Games.	19
Figura 1.8: Promedio de CPI de juegos hiper-casuales en plataforma Android.	21
Figura 1.9: Ranking de Publishers por cantidad de descargas en plataforma.	21
Figura 1.10: Juegos más descargados de App Store en EEUU, Junio de 2018.	22
Figura 1.11: Tiempo usado en juegos móviles en EEUU.	24
Figura 1.12: Caracterización de jugador de dispositivos móviles.	24
Figura 1.13: Chile en la GDC.	25

Capítulo 2: Etapa de pre-producción de un videojuego

Figura 2.0: Flappy bird como referencia para jugabilidad de Astro Kitty.	31
Figura 2.1: Flappy Bird, Mecánica central del juego.	33
Figura 2.2: Jetman, condición de derrota y escenario.	33
Figura 2.3: Progresión de dificultad Astro Kitty.	34
Figura 2.5: Personaje Astro Kitty.	35
Figura 2.5: Movimiento de los repuestos de la nave.	36

Figura 2.6: Flujo de juego de Astro Kitty.	38
Figura 2.7: Vista menú principal Astro Kitty.	39
Figura 2.8: Vista cuenta regresiva Astro Kitty.	39
Figura 2.9: Árbol borroso como referencia a Foreground.	40
Figura 2.10: Midground de Astro Kitty.	40
Figura 2.11: Estructura del escenario de acuerdo a la profundidad de campo.	41
Figura 2.12: Ventana de derrota Astro Kitty.	42
Figura 2.13: Título del juego y logo.	42
Figura 2.14: Astro Kitty viajando por el espacio en su nave.	43
Figura 2.15: Astro Kitty despistado previo al choque con un asteroide.	43
Figura 2.16: Astro Kitty se prepara para un aterrizaje forzado.	44
Figura 2.17: Astro Kitty divisa algo en el horizonte.	44
Figura 2.18: Una cueva con repuestos de nave.	45
Figura 2.19: Fuente Rocco.	46
Figura 2.20: Fuente Bauhaus 93.	46
 Capítulo 3: Etapa de producción de videojuego	
Figura 3.0: Tablero de Trello.	50
Figura 3.1: Cámara perspectiva y cámara ortográfica.	51
Figura 3.2: Elementos correspondientes al núcleo del juego.	52
Figura 3.3: Movimiento de la cueva y relocalización.	53
Figura 3.4: Movimiento vertical de obstáculos.	54
Figura 3.5: Componentes de la UI.	55
Figura 3.6: Tipos de publicidad definidos para Astro Kitty.	56
Figura 3.7: Solución mejorada al Raycast.	57

Figura 3.8: AudioManager en el Inspector de Unity.	57
Figura 3.9: Efectos especiales en repuestos de nave.	58
Figura 3.10: Evolución techo de la caverna.	58
Figura 3.11: Separación de elementos de la cueva al relocalizar.	59

Capítulo 4: Etapa de publicación de videojuego

Figura 4.0: Lógica de instalación de componentes de Android App Bundle.	61
Figura 4.1: Recursos gráficos usados en la publicación de Astro Kitty.	62
Figura 4.2: Sistema de análisis de reseñas de Astro Kitty.	67

Capítulo 5: Conclusiones

Figura 5.0: Evolución del mercado hiper-casual el año 2018.	69
Figura 5.1: Cuota del mercado hiper-casual año 2018.	70
Figura 5.2: Muestra de usuarios por categoría de juego.	71
Figura 5.3: Caracterización del usuario hiper-casual.	71
Figura 5.4: Cantidad de impresiones emitidas por segmento.	72
Figura 5.5: Componentes de un juego y su contraparte en el framework.	73
Figura 5.6: Perspectivas del diseñador y jugador en un videojuego.	74
Figura 5.7: Comparativa del profiler en ambos métodos de instanciación de objetos.	76
Figura 5.8: Botón de información con la política de privacidad de Paperseed.	77
Figura 5.9: Cantidad de Usuarios y Retención D1 de Astro Kitty.	78

Índice de Tablas

Capítulo 4: Etapa de publicación de videojuego

Tabla 4.0: Comentarios de Astro Kitty en Tienda Google Play.	66
--	----

Introducción

El siguiente trabajo busca diseñar e implementar un videojuego Hiper-Casual y su publicación en la tienda digital Play Store de Google. Para ello, se cuenta con un equipo multidisciplinario que utiliza metodologías de desarrollo ágil para la correcta ejecución del proyecto. El objetivo de esta memoria es ser una guía para los futuros desarrolladores que va desde el diseño hasta su desarrollo y publicación. La incorporación de buenas prácticas para crear un producto pulido es un factor determinante en esta industria que cada vez presenta una clientela más exigente.

El contenido de este trabajo se desglosa con el capítulo 1 que estudia el fenómeno de los juegos Hiper Casuales y el mercado de la industria de videojuegos en el mundo, para hacer un hincapié de este género en particular.

Luego, en el capítulo 2 se procede con la etapa de preproducción donde se desarrolla el documento de diseño que permite orientar el trabajo de los colaboradores del proyecto, evitar ambigüedades y fijar el alcance del producto.

En el capítulo 3 se explica la etapa de producción destacando previamente la metodología que se utiliza para el desarrollo del producto, especificando el software de administración de proyectos usado por el equipo de trabajo para posteriormente detallar los elementos que dan la funcionalidad al videojuego.

Ya en el capítulo 4 se menciona sobre el proceso de publicación en la tienda digital de Google, las herramientas de analítica disponibles en el proyecto y la retroalimentación de los usuarios que permiten afinar con actualizaciones el producto ya lanzado.

Finalmente, en el capítulo 5, se presentan las conclusiones obtenidas a lo largo del desarrollo de la aplicación, las nuevas propuestas para afinar el proceso productivo y reflexiones personales del autor.

Capítulo 1: Definición de la oportunidad

1.1 Historia de los videojuegos móviles

Desde el surgimiento de las tiendas digitales, la industria de videojuegos móviles floreció. Sólo App Store de iOS cobija alrededor de 811.911 juegos en julio de 2018 [1], y es sin duda, la categoría dominante dentro del universo de aplicaciones que ofrece la tienda. Según el artículo de PocketGamer [2], esta sorprendente cantidad permite encasillar con líneas borrosas, 3 categorías hasta el año 2017: *Casual games*, *Mid-core games* y *Hard-core games*.

Sin embargo, previo al lanzamiento de estas plataformas digitales el desarrollo de videojuegos era algo completamente distinto. Los juegos se desarrollaban en archivos *.JAR*, y era necesario crear diversas versiones del producto para lograr la compatibilidad en una gama de resoluciones y requerimientos técnicos de los dispositivos, por lo tanto, no existía estandarización alguna tanto para hardware como software. Sumado a que la única forma de monetización era la asociación con grandes industrias que tenían tomado el mercado.



Figura 1.0: Tower Bloxx, juego hiper-casual del año 2005. Fuente: PocketGamer.

El nacimiento de la App Store el 10 de Julio de 2008 y luego Play Store el 22 de octubre del mismo año marcan un hito en el desarrollo de aplicaciones y distribución de contenido digital. En un principio las compras dentro de la aplicación (IAP: *In App Purchases*) no existían, y los juegos casuales dominaban en la tienda: *Angry Birds* es un claro ejemplo de éxito que ya el año 2012 lograba 1000 millones de descargas en ambas plataformas [3].

Con la incorporación de las IAP, se abre un nuevo modelo de negocio: juegos gratis con compras dentro de la aplicación permitían llegar a mayor cantidad de público, siendo *Candy Crush* un ícono de los juegos casuales que adoptan esta forma de monetización y ya en el primer trimestre del año 2013, los jugadores de este juego habían gastado 493 millones de dólares en objetos virtuales [4]. Esta forma de monetización logra desplazar a la dominante que en ese entonces, era la compra del juego. Estos juegos son los pioneros en lo que se define como *Casual Game*, principalmente por una jugabilidad sencilla, fácil de aprender y atraen a una amplia audiencia.

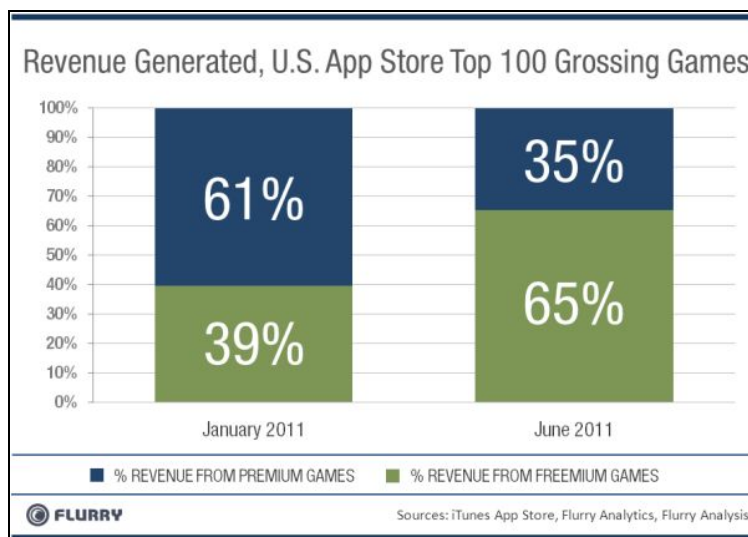


Figura 1.1: 100 juegos con mayor recaudación en EEUU. Fuente Flurry Analytics

A fines del verano de 2012, *Clash of clans* cambió de nuevo los juegos móviles. *Supercell* comprendió que era posible adaptar conceptos de juegos existentes (*Grinding* [5] y *Defensa de la torre* [6] de manera inversa) a una plataforma donde los jugadores experimentan sesiones relativamente cortas. En esta fase, el atractivo *meta-game*¹, los controles adaptados para el tacto y un diseño simple y reconocible llevan los juegos a otros niveles: los *Mid-core games* aparecen en escena. Sumado a esto, ofrecen funciones multijugador y

¹ Meta-game: Conjunto de características del juego que no están directamente relacionadas con la mecánica principal. Su objetivo es mantener a los jugadores comprometidos con la experiencia de juego. Algunos ejemplos son las recompensas diarias por inicio de sesión, conjunto de misiones, logros o cajas de recompensa, entre otros.

sociales asíncronas que dominan las economías virtuales. En términos de distribución, surgieron nuevos canales de marketing escalables y los principales desarrolladores de juegos se enfocan en la adquisición de nuevos usuarios: descubrieron cómo utilizar la promoción cruzada, las redes publicitarias y sociales para encontrar nuevos jugadores. La monetización se convirtió en un híbrido flexible que combina la economía virtual con anuncios de pantalla completa.

Por último están los *Hard-core games*, que se caracterizan por menos sesiones y usuarios, pero un ratio de conversión alto, es decir, pasar de un simple usuario gratuito a alguien que gasta dinero en el juego. Si bien, las líneas entre *Mid-Core* y *Hard-Core* son las más borrosas de todas las categorías, juegos como *Ragnarok Online Mobile* o *Game of War* son claros ejemplos de este tipo. Estos juegos comenzaron a despegar el año 2013, cuando los dispositivos móviles se volvieron omnipresentes y el tiempo dedicado del jugador los obliga a gastar dinero; si alguno no lo hace, está fuera de competencia. Estos juegos son infinitamente escalables, por lo tanto los tratos comerciales deben ser grandes y generosos, y se van volviendo más valiosos hasta que los jugadores se “convierten”. La monetización de estos juegos está enfocada principalmente en los “jugadores ballenas²”: la profundidad mantiene a estos jugadores comprometidos a lo largo de toda la experiencia, que presenta economías virtuales complejas, mecánicas de máquinas expendedoras, incentivos sociales y gestión activa de la comunidad.

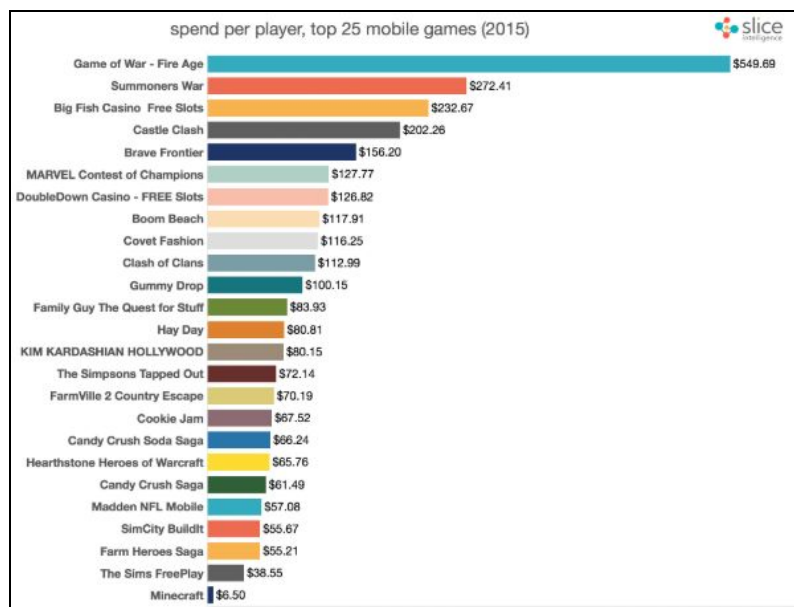


Figura 1.2: Gasto por jugador en el Top 25 juegos móviles el año 2015. Fuente: Slice Intelligence

² Jugadores que gastan mucho dinero en micro-transacciones que se traducen en cosméticos dentro del juego, paquetes de potenciadores, etc.

Juegos como los mencionados anteriormente han sido pioneros en los dispositivos móviles, y las compañías que los han desarrollado forjaron negocios masivos valuados en miles de millones de dólares.

Esta constante evolución del desarrollo de videojuegos móviles dió origen desde el año 2017 a un nuevo género que brilla por su simplicidad y un modelo de negocio muy particular: los juegos hiper-casuales.

1.2 ¿Qué son los juegos hiper-casuales?

Este tipo de juegos siempre ha existido y hasta cierto punto son un resurgimiento nostálgico de los viejos juegos *arcade* de los años 70 y 80, pero solo hace pocos años se convirtieron en un negocio serio. Tienden a atraer a una audiencia amplia, y algunos enganches sociales o incluso modos de multijugador asincrónicos, les ayuda a volverse virales rápidamente, además no demandan ningún compromiso del usuario más que sólo jugar.

Desde la arista visual, los juegos hiper-casuales son simples, incluso minimalistas, generalmente con una interfaz reducida, colores brillantes y formas básicas (cilindros, esferas, etc). La simplicidad de la jugabilidad supone una ventaja para su viralidad: a través de un par de capturas de pantalla, o anuncios cortos los usuarios entienden el juego, reduciendo la barrera de aprendizaje y ayuda a que el producto se descargue y lo jueguen.



Figura 1.3: Frogger (1981) y Crossy Road (2014). Fuente: Elaboración propia.

Este tipo de juegos monetizan a base de anuncios de video con recompensa y sin recompensa, y sólo algunos presentan la opción de un pago por mejoras dentro del juego o

cosmética mediante IAP. Si bien el ingreso es relativamente bajo por usuario activo, esto se compensa por la viralidad, la escala y la rigidez del sistema.

A continuación se analizará el juego *Helix Jump* para describir un flujo de juego sencillo que estuvo número uno en el *ranking* de descargas de las tiendas digitales, liderando con 25 millones de jugadores diarios el último trimestre de 2018.

Juego instantáneo: Una vez instalado el juego es cosa de segundos llegar al *gameplay*. El botón de jugar es muy prominente en el menú principal que incluso hace referencia a una mecánica del juego. No es necesario un tutorial para aprender a jugar, ya que los intentos fallidos van forjando la maestría de una habilidad única y sencilla, hacer “bajar” la bola saltarina por el edificio cilíndrico sin chocar con peldaños que gatillan la condición de derrota. Dependiendo de cuán hábil es el jugador las sesiones duran entre 10 y 180 segundos y si el jugador pierde puede comenzar una nueva en 10 segundos aproximadamente.

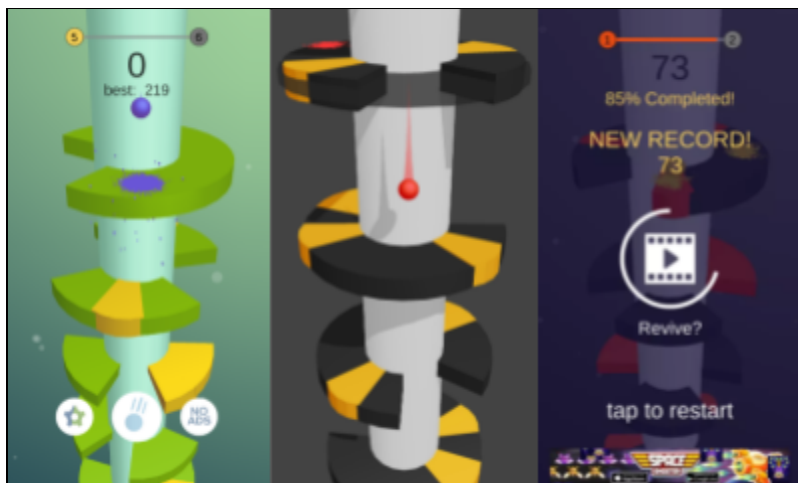


Figura 1.4: Menú principal *Helix Jump*, jugabilidad y sistema de anuncios. Fuente: Elaboración propia.

Monetización:

El sistema de monetización tiene las siguientes características:

1. Video recompensa para obtener una vida adicional antes de comenzar un nuevo juego.
2. Video intersticial después de una cantidad definida de sesiones de juego.
3. Compra de *IAP* para deshacerse de los anuncios publicitarios.
4. Banner publicitario al inferior de la pantalla.

Según PocketGamer [2], se puede definir el siguiente análisis FODA de los juegos Hiper-Casuales:

Fortalezas:

1. Monetización temprana.
2. Bajo CPI³.
3. Bajo costo de desarrollo.
4. Viralidad.

Oportunidades:

1. Mercado enorme.
2. Amplia audiencia.
3. Desarrollo de un portafolio de juegos como compañía.

Debilidades:

1. Períodos de vida cortos.
2. Bajo promedio de ganancias por usuario activo diario (ARPPDAU⁴).
3. Dependiente de una eficiente campaña publicitaria.

Amenazas:

1. Dependiente de la escala.
2. En caso de éxito puede ser copiado rápidamente.
3. Tiende a tener publicidad invasiva.

1.2.1 Principales mecánicas

Como se mencionó anteriormente, estos juegos se destacan por mecánicas muy simples, pero éstas tienen que demandar una concentración total por parte del jugador o detonará la condición de derrota fácilmente. Según *Mobile Free To Play* [8] se pueden mencionar algunas mecánicas que son frecuentes en juegos hiper-casuales:

Mecánica de tap/timing: Es una de las mecánicas más populares de este tipo de juegos y se basa en la exactitud del toque de la pantalla para lograr una acción más eficaz. La precisión es el aspecto más importante de la acción y el enfoque es la perfección. Solo el toque perfecto logrará la mejor puntuación.

³ CPI: *Cost Per Install* es un modelo de precios usado en las campañas de adquisición de usuarios móviles, en el que los anunciantes de aplicaciones pagan cada vez que un usuario instala su aplicación desde su anuncio.

⁴ ARPPDAU: Average Revenue Per Daily Active User.



Figura 1.5: Tiro perfecto en *Baseball Boy*. Fuente: *Mobile Free To Play*.

Mecánica de apilamiento: *Tower Bloxx* (ver figura 1.0) es un juego de apilamiento que además lleva la mecánica de *tap/Timing* necesaria para crear una construcción más elevada. Esta mecánica proporciona puntos de error para los jugadores por la inexactitud del toque en la pantalla y una claridad visual del error de cómo se produjo el fracaso. Mientras menos fracasos, las rondas de juego tienden a ser más largas. En *Tower Bloxx* los puntos de error se expresan al no apilar correctamente los pisos del edificio, haciendo tambalear toda la construcción y dificultando los apilamientos posteriores. El efecto péndulo originado por el balanceo es directamente proporcional por el tamaño del error.

Mecánica de destreza: Estas mecánicas se centran en un jugador que tiene acciones muy simples y repetitivas y se deben realizar cientos de veces. Con suficiente práctica, puede ser dominada por jugadores diestros, por lo que el puntaje más alto es una representación de destreza y habilidad. A medida que el jugador aumenta su destreza, el juego debe tener una dificultad progresiva para que sea divertido. Las acciones deben ser instantáneas y afectarán al personaje inmediatamente. *Flappy Bird* es un ejemplo claro de destreza, aunque no presente una progresión en la dificultad.

Mecánica ascendente/descendente: Esta mecánica provee de viajes interesantes para los jugadores. La progresión constante de nivel conduce a la sensación de progresión sin un cambio en la mecánica o la meta. El enfoque del jugador es lidiar con pequeños desafíos a lo largo del viaje que generan entornos de presión y generan errores autoinflingidos. *Helix Jump* es un juego con una mecánica descendente (ver figura 1.4).

Mecánica de fusión: Esta mecánica radica en que las cosas similares se combinan, y las diferentes no. El juego se vuelve fácil para que las personas lo entiendan bien y con cada fusión subsiguiente, se transmite una nueva comprensión y un fuerte sentido de

progresión, que tiene un carácter exponencial. 2048 es un icónico ejemplo de la mecánica de fusión, que acompañado con una lógica de puzzle tiene como objetivo juntar números pequeños para ir creando más grandes y así llegar al número 2048. Este juego logró un éxito rotundo y fue comparado en muchas ocasiones con *Flappy Bird*, ya que ambos son clones de juegos anteriores. Sin embargo el creador, Gabrielle Cirulli de tan solo 19 años, decidió no monetizar el juego.

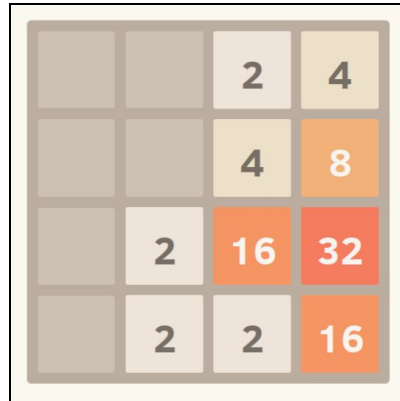


Figura 1.6: Pantalla de juego de 2048. Fuente: Wikipedia.

Mecánica de conducción: Esta mecánica se centra en la conducción usando el arrastre del dedo en la pantalla para evitar los obstáculos. Lo importante de esta mecánica es la precisión de arrastre, en lugar de sincronizar un toque.

Mecánica ociosa: Esta mecánica ha sido usada en *Casual Games* e incluso *Mid-Core* durante varios años, sin embargo la complejidad, el alcance y sobre todo el modo de monetización, define su género y consiste en cualquier progreso del juego que no requiera la participación del jugador. Este tipo de mecánicas se asocian bien con sistemas de economía interna del juego, provocando que los jugadores ganen dinero de manera inactiva y luego lo gasten en la experiencia del juego. Cada segundo que transcurre hace que la probabilidad de pasar algún sub-objetivo sea más fácil, y si se interactúa con el juego se acelera ese instante de tiempo. Sin embargo, cada sub-objetivo siguiente tiene un comportamiento exponencial respecto del anterior, provocando que los segundos necesarios para pasarlo sean más largos. Ahí es donde la monetización con publicidad juega como un engranaje catalizador de sub-objetivos (multiplicando x2, x5 o x10 la cantidad de recompensas obtenidas por segundo es un ejemplo típico).

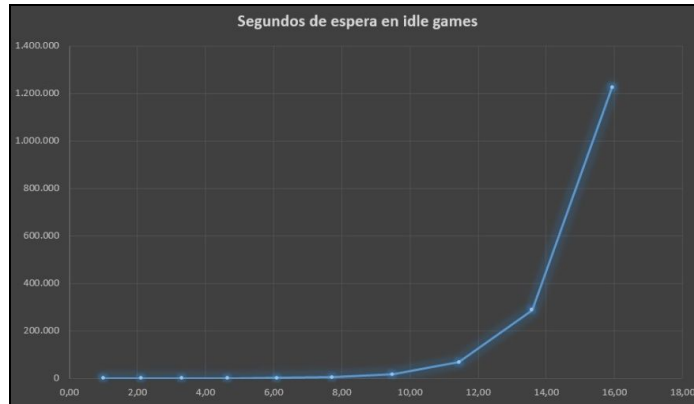


Figura 1.7: Tiempos de espera en Idle Games. Fuente: Nobbot.

Mecánica de crecimiento: Tienen como objetivo central hacer crecer al personaje del juego. Los ganadores de este tipo de juegos son generalmente los más grandes y, en algunos casos, pueden comerse a otros jugadores quienes se encuentran en una arena de combate, siendo una versión muy simplificada de lo que es un *battle royale*⁵. El juego *agar.io* fue precursor de este tipo de mecánicas, en donde los jugadores controlan una o más células de un mapa que representa una placa de Petri. El objetivo es ganar la mayor masa posible comiendo Agar y células más pequeñas mientras se esquivan las de mayor tamaño. Logró un éxito llegando incluso al servicio de streaming *Twitch.tv* y la plataforma de videojuegos Steam.

Mecánica de Puzzle: Los juegos de rompecabezas son un género en sí mismos, sin embargo en el contexto hiper-casual, ésta se centra en la simplicidad en lugar de la complejidad. Un buen juego hiper-casual de rompecabezas, no tiene fin y se centra en las acciones de los usuarios que la progresión de niveles como el juego 2048 mencionado anteriormente (ver figura 1.6).

Si bien, resulta imposible mencionar todas las mecánicas de los juegos hiper-casuales, es notorio cómo algunas compañías toman mecánicas de otros géneros y lo convierten en una experiencia de 1 minuto para el jugador, que sumado a la forma de monetización convierten el juego en hiper-casual.

1.3 Mercado de los juegos móviles

Según Newzoo [9] La industria de juegos móviles ha tenido un crecimiento persistente desde el nacimiento de las plataformas de distribución digital siendo el segmento más grande en el año 2018, ya que reclaman más de la mitad de todos los ingresos globales de juegos por primera vez en la historia: los juegos tanto de celulares como tablets generaron 70.3 mil millones, que representan el 51% del mercado global. Además este segmento tiene la mayoría de jugadores con 2,2 mil millones de usuarios, siendo en gran parte desde

⁵ *Battle royale* es un género de videojuegos que combina los elementos de un juego de supervivencia con la jugabilidad de un último jugador en pie.

teléfonos inteligentes. Los mercados más grandes son el Chino con una “población en línea” de 850 millones de personas y ganancias que bordean los 38 mil millones, seguido de Estados Unidos con una población respectiva de 265 millones y ganancias de 30 mil millones.

Si bien, los juegos con mayor recaudación son los que presentan el sistema IAP, como *Fortnite* o *Candy Crush*, la forma de monetización más usada por los desarrolladores es la publicidad con un 58% de los juegos existentes (AppFigures, 2016). Esto se debe principalmente a que el 54% de jugadores móviles prefieren ver anuncios de video si esto lleva recompensas dentro del juego (Gamasutra, 2016).

En el año 2017, el alcance del mercado de los juegos más populares de Android sitúa al género casual en la primera posición con un ratio de penetración de un 59.6%, siguiendo la categoría rompecabezas con un 57.26% y, en tercer lugar, los arcade con un 55.6%. Estos géneros son ideales para ser monetizados de una manera hiper-casual (Shared2you, Septiembre 2017).

De la perspectiva de los desafíos de la industria, 73.1% de los desarrolladores considera que la competencia en el mercado es una de las mayores amenazas para el negocio; en segundo lugar, se encuentra el aumento en los costos de adquisición de usuarios con un 63.2% y con un 47.5% es preocupante para la industria el aumento de los costos en *marketing* (mobyaffiliates, Mobvista y PocketGamer.biz el 2017). En este sentido, los juegos hiper-casuales presentan una ventaja competitiva: mientras que el costo de instalación de un juego con un alcance mayor ronda los 2.05 USD, los juegos hiper-casuales pueden lograrlo por un valor 7 hasta 13 veces inferior [10].

Si además se consideran las variables de plataforma de publicación y el porcentaje del total de instalaciones por país, se concluye que Latinoamérica e India ofrecen oportunidades económicas de crecimiento a través de Facebook, colocando a Brasil en una situación favorable para iniciar lanzamientos de videojuegos [10].

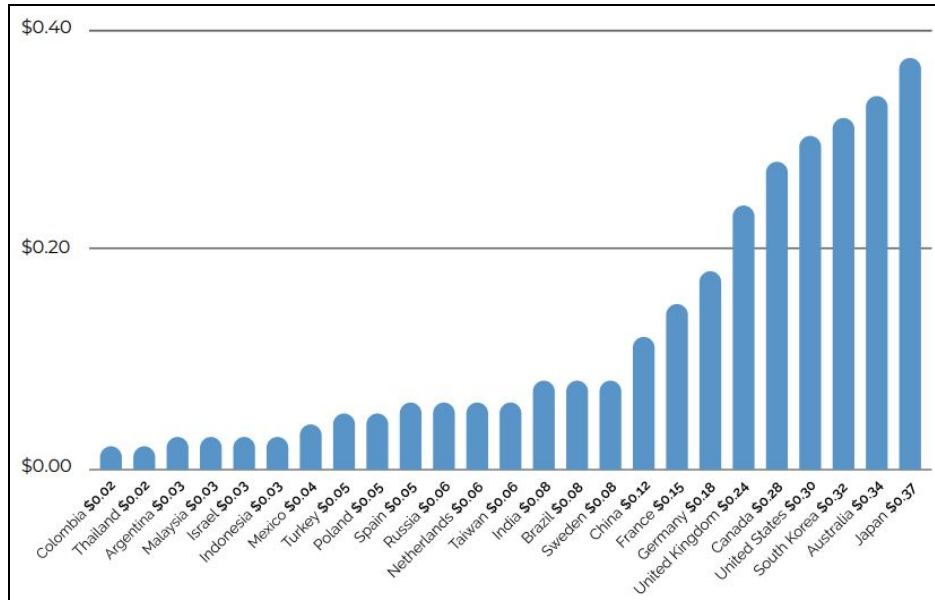


Figura 1.8: Promedio de CPI de juegos hiper-casuales en plataforma Android. Fuente: Tenjin.

Con respecto a los juegos infantiles, el 69% de las ganancias es generada por juegos de pago, sin embargo se espera un crecimiento saludable para los juegos gratuitos sobre todo por ser la plataforma más popular en este segmento de la población, seguido de las consolas (SuperData Research, 2015). Alrededor del 37% de los consumidores de consolas de videojuegos asegura jugar una versión móvil de los juegos si estuviera disponible en el mercado (AYTM, 2016).

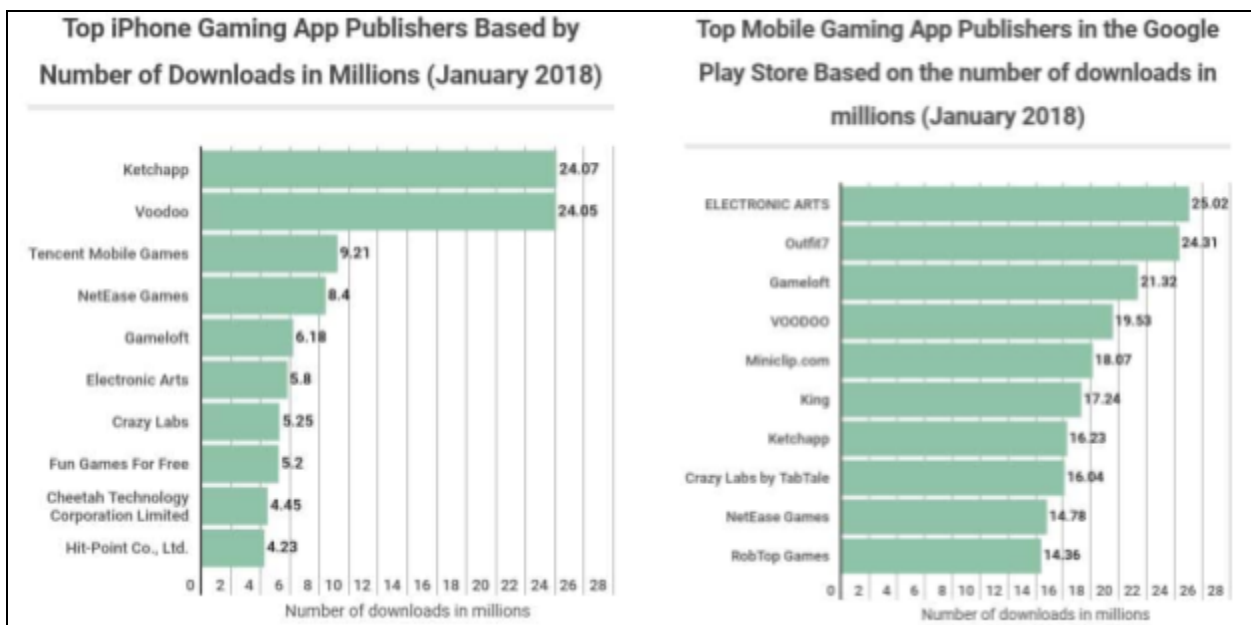


Figura 1.9: Ranking de Publishers por cantidad de descargas en plataforma. Fuente: Priori Data.

1.3.1 Auge de los juegos Hiper-Casuales en el mercado

Los juegos hiper-casuales han experimentado un crecimiento sin precedentes en el mercado. Usando el promedio de vida de un jugador hiper-casual, IronSource estima que el mercado aproximado de este género se ubica entre los 2 mil millones a 2.5 mil millones de dólares. Y es que para los jugadores casuales, incluso descargar una aplicación, es a veces una barrera para jugar. Según Newzoo [9], desde 2017 que las plataformas de redes sociales están experimentando con juegos que se puedan jugar al instante, sin necesidad de descargarlos. Facebook fue uno de los precursores, que se abrió a desarrolladores externos en Marzo de 2018, y Google lo siguió rápidamente. La práctica ya es un lugar común entre las redes sociales en los países asiáticos, donde Tencent en China, Kakao en Corea del Sur y LINE en Japón, ofrecen experiencias de juego integradas instantáneamente directamente de sus aplicaciones. La tendencia es beneficiosa para todas las partes involucradas: mayor cantidad de usuarios en la aplicación y *marketing* en su máxima expresión. Los jugadores ocasionales pueden probar los juegos inmediatamente y la integración con la red permite una configuración rápida y fácil para juegos sociales. Los desarrolladores se exponen a la enorme bases de usuarios de las plataformas, mientras que las redes sociales se benefician de un mayor compromiso en su sistema y obtienen una parte de cada transacción en estos juegos. Este último beneficio es crucial, ya que permite a estas redes sociales cumplir con el rol de distribuidor previamente reservado para las tiendas de aplicaciones. Sin embargo, estos juegos tomaron mayor visibilidad cuando la empresa *Ketchapp* (ahora propiedad de *Ubisoft*) entró en el mercado, creando un portafolio rápido de productos que están interconectados. Incluso Voodoo, con una docena de juegos y 100 millones de instalaciones el año 2017, recibió un fondo de inversión de 200 millones de dólares de *Goldman Sachs*, que les permitió consolidar el trono de los juegos hiper-casuales[11].

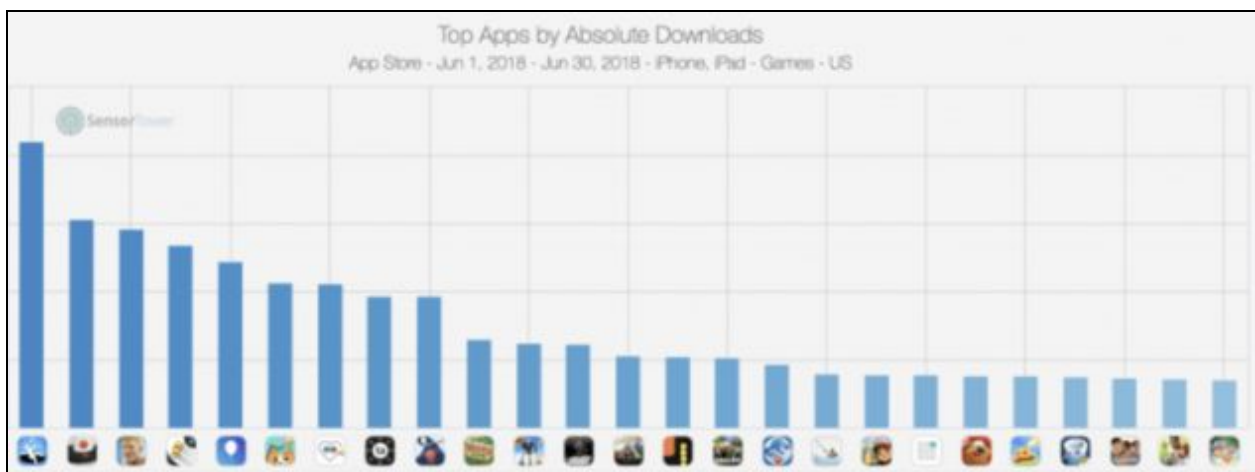


Figura 1.10: Juegos más descargados de App Store en EEUU, Junio de 2018. Fuente: Sensor Tower.

Al analizar los datos estimados de *Sensor Tower* (figura 1.10), *Voodoo* representó el 24.7% de todas las descargas gratuitas, ahora si se amplía la vista a todas las empresas de este género esta cifra aumenta hasta un 57%: ningún otro género controla la cantidad de descargas como lo hacen los juegos hiper-casuales. Y es que la simplicidad del juego, junto con la velocidad de aprendizaje de los jugadores crea una necesidad voraz de descargar la próxima nueva idea, y los juegos antiguos o que no lograron cautivar son descartados de ser publicitados o se eliminan rápidamente. *Voodoo* puede tener 100 o 200 estudios de desarrollo que trabajan frenéticamente en nuevos diseños, poniendo el foco en la mecánica central. Incluso los objetivos principales o la publicidad son omitidos para el primer lanzamiento de *software*. Estos bucles de testeo se prueban en territorios clave como EEUU o China para ver si existen buenas respuestas de la mecánica central. La receta de *Voodoo* [12] es:

Snackable: Hace mención a que el contenido sea fácil de digerir, con experiencias de 2 minutos como máximo. Además se pueden jugar en cualquier lugar y momento.

Youtubable: Contenido sencillo de entender con un video o GIF.

Directo hacia adelante: No es necesario un tutorial para entender el juego.

No castiga: Da la sensación de libertad y progresión al jugador, además de hacerlo sentir exitoso en el juego.

Innovador: Crear una mecánica no existente en el mercado, combinando las existentes.

Esta forma de lanzar productos es muy distinta y bastante más ágil que los juegos tradicionales.

Estos juegos se miden en una escala muy dura. Cada juego necesita una retención de usuarios en el día 1 de más de un 50% [13]. Esto crea un entorno competitivo donde las estadísticas y los datos se convierten en la clave para ser recogidos por los publishers.

En un sistema donde el CPI es bajo, los juegos pueden crecer rápidamente. Si el *LTV*⁶ es mayor al CPI, entonces el objetivo es seguir gastando para crecer aún más rápido. Y por ello el hiper-casual funciona tan bien, porque puede reducir el CPI a niveles increíblemente bajos a través de una publicidad clara y un *marketing* impulsado por datos (hiper-segmentación de clientes).

1.3.2 Caracterización del usuario que descarga juegos hiper-casuales

Para comprender el efecto de los juegos hiper-casuales en la industria, es necesario analizar cómo este género tuvo un crecimiento tan rápido: Hoy en día los los juegos

⁶ *LTV: Lifetime Value* es un indicador que determina cuánta ganancia genera un usuario durante toda su vida útil dentro de una aplicación.

móviles son más populares entre las mujeres, quienes representan al 63% del mercado, además un 38% de la población *gamer* tiene más de 45 años [14].



Figura 1.11: Tiempo usado en juegos móviles en EEUU. Fuente: Mediakix.

Según el informe EEDAR [15], la forma en que las personas juegan también está cambiando: en lugar de largas sesiones de juego las personas prefieren hacerlo realizando múltiples tareas en la casa, seguido de “esperando a alguien”, mientras viajan, toman un descanso o van al baño. En todas estas situaciones, los jugadores buscan cada vez más entretenimiento de bajo compromiso y que puedan disfrutar en ráfagas cortas, en consecuencia el hiper-casual encaja bien con esta tendencia.

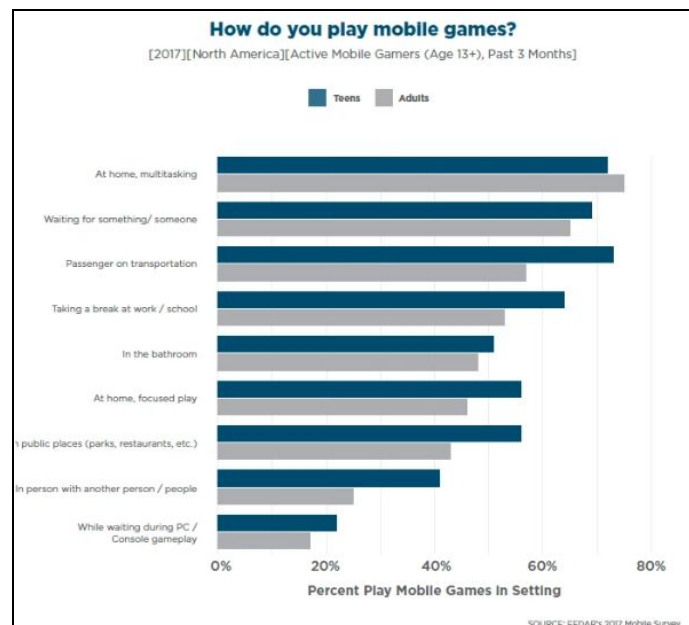


Figura 1.12: Caracterización de jugador de dispositivos móviles. Fuente: EEDAR 2017.

1.3.3 Industria de videojuegos en Chile

La industria de videojuegos en Chile ha tenido altos y bajos. Hace 10 años, si los emprendimientos apenas se lograban sostener, en la actualidad los estudios independientes proliferan e incluso logran importantes fondos de inversión en el extranjero, como es el caso de TinyBytes Games, que recibió un fondo internacional de 1.4 millones de dólares.

Los inicios de la industria se remontan a los años 90 cuando en el mercado existían alrededor de 5 empresas independientes. Sin embargo, no fue hasta el 2007 la idea de establecer un gremio de desarrolladores. Así nace VideoGames Chile que en el año 2010 pudo constituirse legalmente, ya que no contaba con los miembros suficientes para fundarlo. Hoy esta organización agrupa el 75% de los 60 estudios del país [16].

El punto de inflexión de la industria fue la relación con ProChile, organización estatal encargada de promocionar productos nacionales en el extranjero, que en el año 2012 envió a la primera delegación de industria en Iberoamérica en estar presente en la Game Developers Conference en San Francisco. Actualmente las empresas chilenas visitan las ferias de videojuegos más importantes del mundo, teniendo la posibilidad de realizar contactos con inversionistas y publishers de todas las plataformas.



Figura 1.13: Chile en la GDC. Fuente: ProChile.

Sin embargo, grandes empresas han dejado de operar en Chile, como fue el caso de la empresa japonesa DeNa y Behaviour de Canadá. Con la salida de estas 2 empresas la industria chilena pasó de tener más de 12 millones de dólares de facturación en 2016 a recaudar cerca de 6 millones de dólares el 2017.

Actualmente la empresa más grande del rubro en el país es Gamaga. Cuenta con 30 empleados aproximadamente y desarrollan videojuegos móviles. Además es la primera empresa chilena en romper la facturación del millón de dólares en menos de un año.

En conversaciones con Andrés Sepúlveda, quien es fundador de Inner Child, empresa de Viña del Mar dedicada al desarrollo de videojuegos, comenta que en sus inicios recibió fondos estatales para el desarrollo de experiencias Web, pero deciden en el último mes de financiamiento cambiarse a los hiper-casuales. Han desarrollado más de 20 juegos y actualmente tienen un contrato con Voodoo para el prototipado de juegos.

1.4 Equipo de Trabajo

El desarrollo del videojuego presentado en esta memoria será un producto de Paperseed, una micro-empresa emergente de videojuegos y experiencias en AR/VR de Viña del Mar, cuyo objetivo es darse a conocer con este primer proyecto llamado Astro Kitty. La empresa está formada por un Programador, Modelador/Animador, Ilustradora e Ingeniera y diseñadora de sonidos. La experiencia del autor de esta memoria diseñando pequeños videojuegos en diversos Workshop, le permite estar en una posición sólida para orientar y guiar al equipo en el trabajo conjunto y lograr el desarrollo integral del proyecto.

1.5 Objetivos de la memoria

1.5.1 Objetivo Principal

El siguiente trabajo busca crear un videojuego del tipo hiper-casual desde su diseño utilizando un documento estandarizado como GDD (Game Design Document), su desarrollo e implementación en el software Unity 3D y su publicación en la tienda de Google, a modo de recopilar un conjunto de buenas prácticas que sirvan para sistematizar futuros procesos de desarrollo de la empresa Paperseed creando productos de alta calidad para esta industria que cada vez presenta usuarios más exigentes.

1.5.2 Objetivos Específicos

1. Analizar el mercado actual de la industria, con mención en la categoría juegos “hiper-casuales”. Esto permitirá clarificar la idea de negocio para luego implementar una monetización correcta en este tipo de videojuegos.
2. Creación de un documento de diseño de videojuegos: siendo una de las etapas claves en el proceso de producción ya que da una idea clara a todo el equipo de producción de los elementos que compondrán el juego.
3. Desarrollo de software usando la herramienta Unity: identificando las diferentes mecánicas requeridas y siguiendo las etapas de producción de videojuegos según la metodología KANBAN y su publicación en Google Play.

4. Sondeo estadístico de la experiencia de usuario para generar mayor claridad al momento de implementar la publicidad.

Capítulo 2: Etapa de pre-producción de un videojuego

2.1 Introducción

La etapa de pre-producción es clave, pues acá se realizan las decisiones más importantes del diseño: la mecánica de la jugabilidad ya es validada por el equipo mediante un prototipo, se define el alcance del proyecto, estilo artístico, requerimientos técnicos y la calendarización de producción, todo esto empaquetado en un documento de diseño.

Su objetivo principal es proveer una descripción detallada de cada aspecto que necesita ser realizado, actuando así como una enciclopedia del juego, donde el equipo puede realizar un seguimiento de qué se ha hecho y cómo.

Tal como lo expone el libro *Practical Game Design* [17], las características fundamentales de este documento son:

Modular: Organizar el GDD en módulos de manera tal que las funcionalidades se manejen de manera independiente es imprescindible para mantener el orden del proyecto, la actualización del documento mismo, el manejo de dependencias y la reusabilidad para futuros proyectos.

Definición de objetivos y requerimientos: No solo para el proyecto en general, sino que para cada módulo del juego.

Es el resultado de una discusión: Una vez definido los objetivos y requerimientos es necesario socializar con el equipo las características del juego, siendo los programadores los interlocutores más importantes, ya que ellos serán los que implementan el diseño. El uso de lluvia de ideas es excelente para sistematizar esta etapa. El objetivo principal de esta sección es evitar ambigüedades llegado el momento de la producción del videojuego.

Es claro y conciso: El GDD debe ser leído por el equipo de trabajo, por eso si es demasiado extenso e implica horas de estudio, puede ser contraproducente. Si una característica del videojuego toma una docena de páginas, es mejor dividirla en sub-características.

Multimedial: Para favorecer el punto anterior es indispensable recopilar referencias de imágenes o videos que permitan mejorar la cohesión del proyecto. Una imagen vale más que mil palabras y un video más que mil imágenes.

Deja espacio para la creatividad y el debate: La idea central es comunicar la intención más que la implementación. Existe la posibilidad de que se planteen inquietudes o preguntas sobre lo que hay escrito y éstas deben abordarse para finalmente crear una nueva iteración del documento. Aún más probable, estos problemas pueden surgir ya en la etapa de desarrollo, lo que implica una rápida solución sobre la marcha, o incluso, que la implementación real diverge del GDD pero se apega más a la visión del juego y sea mejor que el diseño original. En ese caso, es el GDD el que debe ser modificado y no la implementación.

Diferentes formatos y tamaños: Las herramientas de ofimática de Google u Office son excelentes para documentar el proceso de pre-producción, siendo el Documento de Texto y las presentaciones los principales. Para juegos donde escala en lo numérico una hoja de cálculo es fundamental.

Este documento de diseño tiene como objetivo principal plasmar todos los elementos que se deben incluir en Astro Kitty, además de servir de carta de presentación para buscar colaboradores futuros. A continuación se detalla el contenido del GDD para Astro Kitty.

2.1.1 Concepto del juego

Astro Kitty es un videojuego hiper-casual para celulares donde el usuario controla un gato espacial que se ha estrellado en un planeta desconocido y tiene como objetivo encontrar en una cueva repuestos para su nave. Para ello deberá sortear las dificultades que le presenta la caverna, saltando y esquivando obstáculos.

2.1.2 Características principales

El juego se basa en los siguientes pilares:

Planteamiento simple: la historia mencionada es muy simple, sin embargo lo suficientemente explícita para que el jugador tenga un objetivo claro durante todo el transcurso del juego.

Destreza: Para esquivar los obstáculos de la caverna el jugador deberá interiorizarse con el movimiento del gato ya que cualquier *tap* en falso puede ocasionar perder en el juego.

Concentración: El juego demanda al jugador constante atención en lo que dure una sesión de juego.

Nivel re-jugable: El juego consta de 1 nivel que se genera en función de los repuestos de nave que va capturando el jugador, haciendo cada vez más difícil sortear los obstáculos. Cada partida es distinta en cuanto al escenario.

2.1.3 Género

Astro Kitty, entra en la categoría de juego hiper-casual siendo la mecánica central un *Endless Runner*: El jugador debe avanzar de manera irremediable en una misma dirección, generalmente escapando de algún enemigo o peligro, y cuyo objetivo es avanzar lo máximo posible antes de morir.

2.1.4 Público dirigido, propósito y requerimientos

El juego está dirigido a jugadores de dispositivos móviles, que enganchan rápidamente con la jugabilidad instantánea, quieren desafíos y sientan una motivación en el juego por aprender y mejorar sus habilidades en el tiempo, siguiendo la premisa “fácil de aprender, difícil de dominar”. Rango etario va desde 12 a 45 años.

El propósito principal del videojuego es ofrecer una experiencia de un solo jugador que rememora la famosa mecánica de *Flappy Bird*.

Los principales requerimientos del proyecto son lograr un diseño artístico cautivador, ser un juego liviano y periódicamente reciba actualizaciones para lograr el objetivo de la historia del videojuego.

2.1.5 Jugabilidad

Como buen *Endless Runner*, cada sesión de juego posee los mismos elementos, sin embargo es distinta en cuanto a disposición, generando una infinidad de escenarios. A continuación se detallan algunos elementos esenciales:

Tap para saltar: El jugador debe estar saltando constantemente para mantenerse en la ruta y no chocar con los obstáculos, que son la caverna en sí y las barreras que aparecen cada cierto tiempo.

Puntuación: El jugador debe capturar los repuestos que aparecen aleatoriamente en el escenario para aumentar su puntaje.

“Vida extra”: El jugador tendrá la opción de acceder a una “vida extra” luego de haber colisionado con algún obstáculo.

Orientación de celular: “Portrait”

Una buena referencia visual de la jugabilidad es la que se aprecia en el siguiente video del juego *Flappy Bird*: <https://www.youtube.com/watch?v=n6xa8pxe1cE>

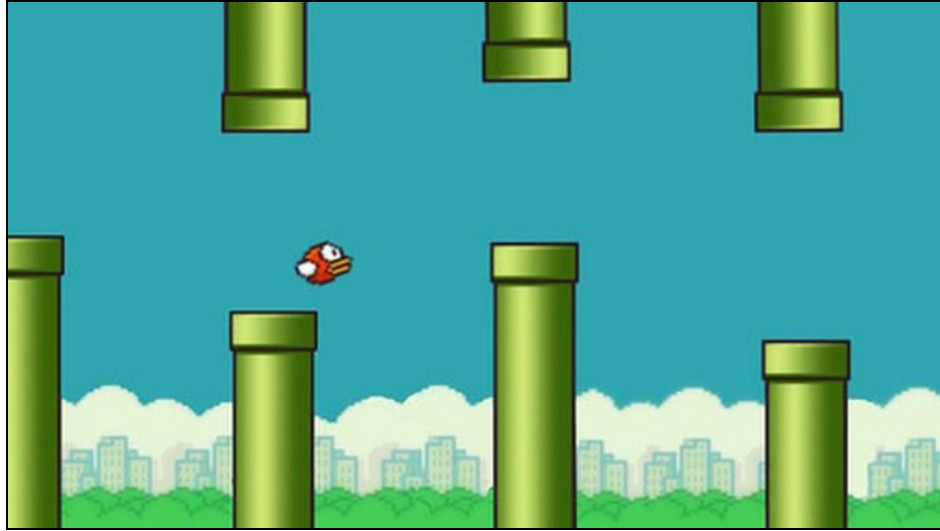


Figura 2.0: Flappy bird como referencia para jugabilidad de Astro Kitty. Fuente: Flappy Bird.

2.1.6 Alcance

- Experiencia de juego sencilla pero altamente repetible.
- Flujo de juego completo.
- Progresión de dificultad.
- Implementación de *Unity Ads*.
- Publicación en *Google Play*.

2.2 Especificaciones

2.2.1 Plataforma

Astro Kitty es un juego para dispositivos móviles con sistema operativo *Android*.

Nivel mínimo de API de *Google* para funcionamiento del juego: 4.1 *Jelly Bean*.

Arquitecturas de celular: ARMv7, ARM64, x86.

2.2.2 Requerimientos técnicos

Motor de videojuego: Unity versión 2018.3.13f1.

Requerimientos de sistema:

- OS: Windows 7 SP1+, 8, 10, 64-bit.
- CPU: Soporte para el conjunto de instrucciones SSE2.
- GPU: Tarjeta de video con capacidad para DX10 (*shader model* 4.0).
- Requisitos adicionales para el desarrollo de plataforma android:
- Android: *Kit* de desarrollo Android SDK y Java (JDK); el IL2CPP *scripting backend* requiere Android NDK.
- Plataforma Windows universal: Windows 10 (64 bits), Visual Studio 2015 con componente C++ Tools o posterior y SDK para Windows 10.

IDE: Visual Studio Code.

Ilustración: Adobe Photoshop CS6.

Requerimientos de sistema:

- Procesador Intel® Core 2 o AMD Athlon® 64 (2 GHz o más rápido).
- Microsoft Windows 7 con Service Pack 1, Windows 8.1 o Windows 10 (versión 1607 o posterior)
- 2 GB o más de RAM (se recomiendan 8 GB)
- Pantalla de 1024 x 768 (se recomienda 1280 x 800) con color de 16 bits y 512 MB o más de VRAM dedicado; se recomiendan 2 GB*
- Sistema compatible con OpenGL 2.0

Animación 2D: Moho Pro 12.

Requerimientos de sistema:

- Windows® 7, 8.1, 10.
- Sistema operativo 64-bit.
- 1.3 GHz Pentium 4 mínimo.
- 2 GB RAM recomendado.
- 1.6 GB espacio mínimo en disco duro.
- OpenGL habilitado GPU recomendado
- 1440x900 Densidad de pantalla (1920x1080 recomendada)

Musicalización y efectos especiales: Pro Tools 11 y Studio One.

Requerimientos de sistema:

- Windows 7, Windows 8.1 o Windows 10 x64
- Procesador Intel® Core™ Duo o AMD® Athlon™ X2 (Intel Core 2 Duo o AMD Athlon X4 ó superior recomendado)
- 4 GB RAM mínimo (8 GB ó más recomendados)
- Monitor con resolución de 1366 x 768 (se recomienda monitor con un alto rango de ppp)
- 40 GB de espacio en el disco duro

2.3 Mecánicas y componentes de juego

Esta sección detalla todos los pilares que fundamentan la jugabilidad de Astro Kitty y las acciones que lleva a cabo un jugador en una partida típica.

2.3.1 Jugador

Físicas: El gato está sujeto a fenómenos físicos, con gravedad simulada a la tierra. La física sólo está activa al momento de apretar el botón de *play*.

Salto: El jugador puede saltar para esquivar los obstáculos haciendo *tap* en el celular, siendo este $\frac{1}{6}$ de la altura del escenario y no tiene tiempo de enfriamiento. Es el único movimiento que tiene el jugador.

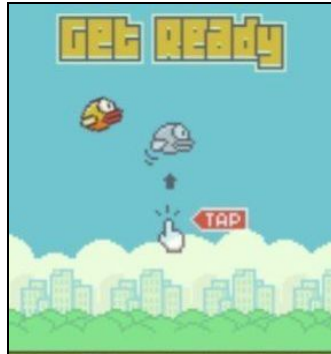


Figura 2.1: Flappy Bird, Mecánica central del juego. Fuente: Flappy Bird.

Puntuación: El jugador obtiene puntos atrapando “repuestos de nave” que aparecen en el escenario. Cada repuesto atrapado suma 1 al puntaje de la partida. Cuando el jugador pierde, el puntaje tiene 2 caminos:

- Ser reseteado apretando el botón de reiniciar.
- Mantener el puntaje apretando el botón de “vida extra”.

Condición de derrota: El jugador pierde colisionando con la caverna: techo, suelo y obstáculos de cristal.



Figura 2.2: Jetman, condición de derrota y escenario. Fuente: Youtube.

“Vida extra”: El jugador puede obtener una “vida extra” presionando el botón luego de efectuarse la condición de derrota.

Mejor Puntaje: El sistema guarda el mejor puntaje del jugador.

2.3.2 Escenario

El escenario jugable se compone de techo, suelo, repuestos de nave y obstáculos de cristal.

Reproductor de elementos: Tanto los obstáculos como los repuestos de nave aparecerán con una frecuencia determinada para generar la dinámica de “esquivar y atrapar”. El reproductor de elementos considera para los obstáculos la progresión de dificultad y para los repuestos de nave una aleatoriedad simple.

Progresión de dificultad: Cuando el jugador supera una cantidad de puntaje, el espacio entre el obstáculo y el techo se angosta:

- Puntaje menor a 25 dificultad fácil
- Puntaje entre 25 a 45 dificultad media-fácil
- Puntaje entre 45 a 65 dificultad media
- Puntaje entre 65 a 85 dificultad media-difícil
- Puntaje mayor a 85 dificultad difícil

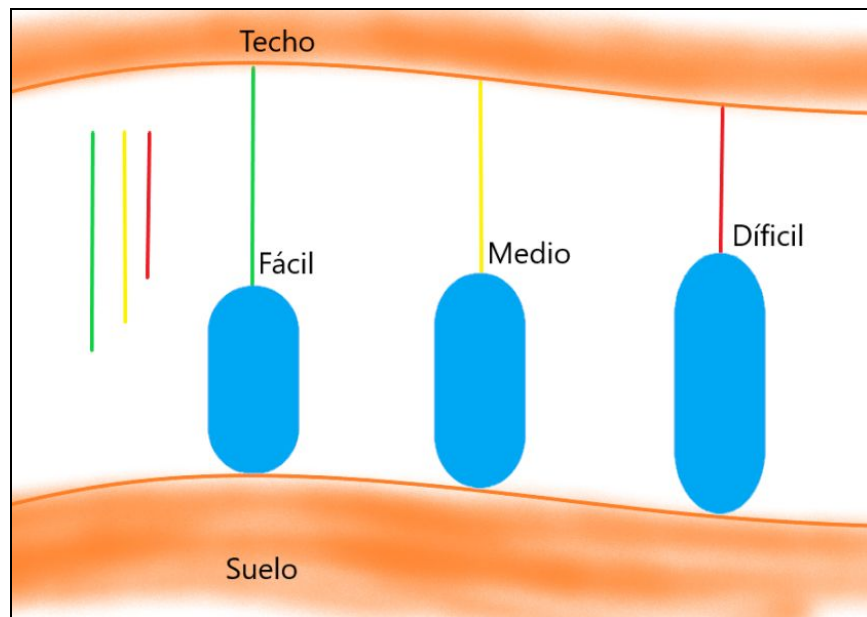


Figura 2.3: Progresión de dificultad Astro Kitty. Fuente: Elaboración propia.

2.3.3 Personaje

Astro Kitty es un gato espacial que se encuentra volando en su nave para una misión interplanetaria. Cuando un cometa pasa cerca de él, se distrae y es colisionado por un asteroide perdiendo el control de la nave. Afortunadamente logra aterrizar en un planeta que si bien es inhóspito, tiene una caverna llena de repuestos para naves. Su misión ha cambiado y tiene como objetivo principal encontrar las piezas que le permitirán reparar su nave para salir de ese lugar.



Figura 2.4: Personaje Astro Kitty. Fuente: Paperseed.

2.3.4 Control del jugador

El control del jugador es únicamente el *tap* del celular que le permite usar de la mecánica de salto de Astro Kitty, además de navegar por la interfaz de usuario.

2.3.5 Movimientos del escenario

Movimiento caverna: Se mueve horizontalmente teniendo distintas velocidades dependiendo de la profundidad de campo.

- Estrellas: Velocidad lenta
- Fondo de la cueva: Velocidad media-lenta
- Piso, techo y obstáculos: Velocidad media
- Frente de la cueva: Velocidad media-rápida

El movimiento se detiene al perder el jugador; y al apretar el botón “reiniciar” o “vida extra”, vuelven a su posición inicial.

Los obstáculos además varían en su altitud dependiendo del techo generando un espacio acotado entre estos elementos, permitiendo al jugador poder pasar entre ellos.

Movimiento “repuestos de nave”: Se mueve horizontalmente, pero siguiendo una trayectoria sinusoidal. Posee la misma velocidad que el piso, techo y obstáculos. El movimiento se detiene cuando el jugador pierde; y al apretar el botón de “volver a jugar” o “vida extra”, vuelven a su posición inicial.

Cada tipo de repuesto tendrá una amplitud y frecuencia determinada.

Una buena referencia del movimiento que se desea lograr se encuentra en el siguiente video: <https://www.youtube.com/watch?v=x5f4zgYxZHc>

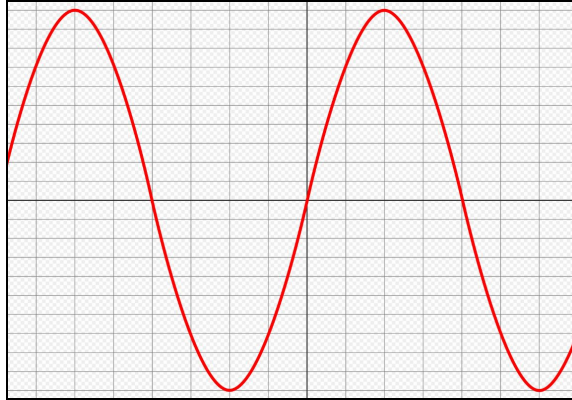


Figura 2.5: Movimiento de los repuestos de la nave. Fuente: Wikipedia.

Movimiento techo caverna: Luego de un tiempo, el techo de la caverna varía en altura de forma aleatoria. Vuelve a su posición original después de perder.

2.3.6 Interacción entre elementos

Astro Kitty presenta las siguientes interacciones entre elementos:

Personaje:

- Techo de caverna: El jugador colisiona con el techo, entonces pierde.
- Suelo de caverna: El jugador colisiona con el suelo, entonces pierde.
- Obstáculos caverna: El jugador colisiona con los obstáculos, entonces pierde.
- Repuestos de nave: El jugador colisiona con los repuestos de nave, entonces aumenta su puntaje y se destruye el objeto repuesto de nave.

Reproductor de elementos:

- Obstáculos: Crea el objeto obstáculo.
- Repuestos de nave: Crea el objeto repuesto de nave.
- Techo: Mantienen una distancia fija entre ellos.

Repuestos de nave:

- Fin del mundo: El repuesto de nave colisiona con objeto “fin del mundo”, entonces repuesto de nave se destruye.
- Estela: La “estela” sigue la trayectoria de los repuestos de nave

Control de jugador:

- Botón de play: Inicia el juego.
- Botón de opciones: Abre el panel de opciones.
- Botón de sonido: Activa/Desactiva sonidos de efectos especiales.
- Botón de música: Activa/Desactiva música del juego.
- Personaje: Ejecuta salto del personaje.
- Botón de vidaextra: Despliega publicidad recompensa.
- Botón de saltar: Cambia de escena para llegar al menú principal.

- Botón de reiniciar: Volver al menú principal.
- Botón de saltar publicidad: Salta la publicidad.

2.4 Interfaz

Astro Kitty destaca por su interfaz sencilla y minimalista. A continuación se detalla el flujo del juego y cada pantalla que compone la experiencia completa del jugador.

2.4.1 Flujo de juego

El jugador inicia Astro Kitty reproduciendo un pequeño video que explica el argumento del juego, y la primera vez que lo inicia éste no se puede “saltar”. Finalizado el video se accede al menú principal que se encuentra en la misma escena de la jugabilidad. Desde este menú es posible acceder al botón de configuraciones que despliega hacia abajo una serie de opciones activables para controlar los sonidos de efectos especiales, la música y un link a la página web de Paperseed con la política de privacidad; y el botón de jugar que al ser apretado desactiva la interfaz del menú e inicia una cuenta regresiva de cuando da inicio al juego. Finalizado ese estado se procede con la jugabilidad en sí. Cuando el jugador pierda en dicha partida, se accede al menú de derrota, que presenta dos botones: el primero es un Game Over tradicional que manda al jugador al menú principal, pero que cada 3 derrotas despliega el sistema de monetización de Unity con publicidad intersticial con posibilidad de “saltar”, y el siguiente permite obtener una “vida extra” viendo una publicidad recompensa de 30 segundos sin posibilidad de “saltar”, para luego volver al menú principal con el puntaje de la partida recién pasada.

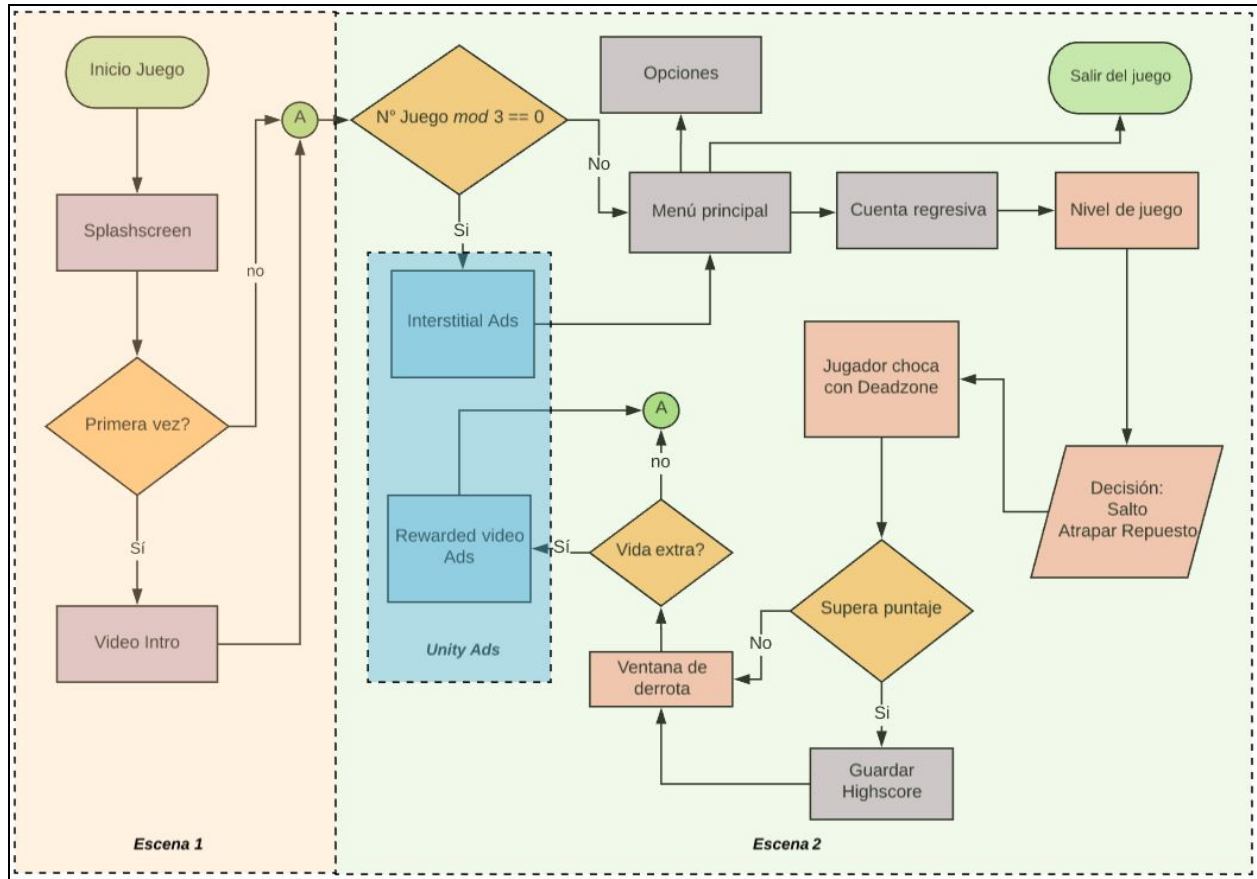


Figura 2.6: Flujo de juego de Astro Kitty. Fuente: Elaboración propia.

2.4.1.1 Menú Principal

El menú principal (ver figura 2.7) consta de las siguientes funciones:

Play: Dar inicio al juego.

Opciones:

- Activar/Desactivar Sonidos de efectos especiales.
- Activar/Desactivar Música.
- Link a página web con la política de privacidad de la empresa.

Salir: Cerrar aplicación.



Figura 2.7: Vista menú principal Astro Kitty. Fuente: Elaboración propia.

2.4.1.2 Ventana de cuenta regresiva

Estado de transición entre el menú principal y el nivel de juego, que es donde se efectúa el *gameplay*.

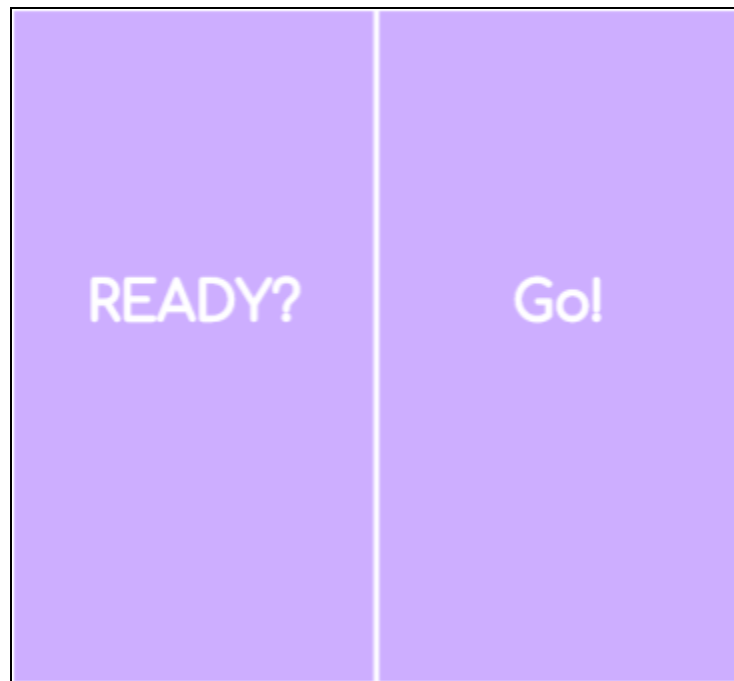


Figura 2.8: Vista cuenta regresiva Astro Kitty. Fuente: Elaboración propia.

2.4.1.3 Nivel

El escenario de juego se separa en capas de acuerdo a la profundidad de campo. A continuación se detallan los elementos de acuerdo a la estructura señalada.

Foreground: Puramente estético.

- Pilares de caverna: tono más oscuro y efecto de borroso para destacar que es un plano no jugable.



Figura 2.9: *Árbol borroso como referencia a Foreground. Fuente: Limbo.*

Midground: Plano jugable.

- Techo de caverna
- Suelo de caverna
- Obstáculos
- Repuestos de nave

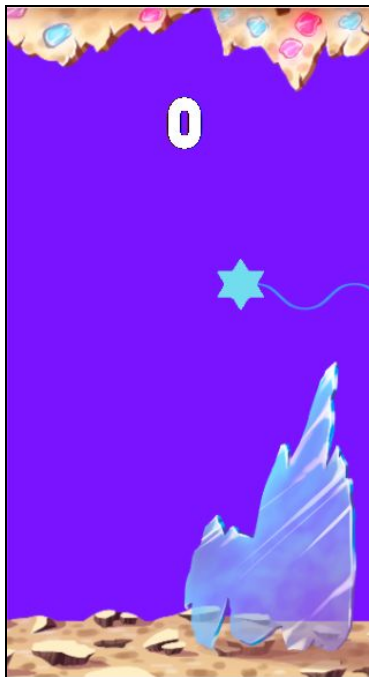


Figura 2.10: *Midground de Astro Kitty. Fuente: Elaboración propia.*

Background: Plano estético

- Fondo de la cueva
- Estrellas
- Skybox

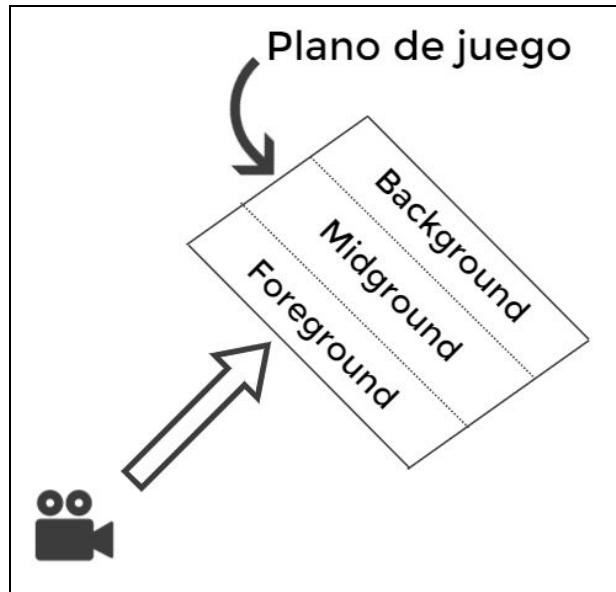


Figura 2.11: Estructura del escenario de acuerdo a la profundidad de campo. Fuente: Elaboración propia.

2.4.1.4 Splash Screen

Se utiliza el Splash Screen por defecto de Unity Free. Consiste en la presentación en 2 segundos de el logo de la empresa (Paperseed) y “Made with Unity”.

2.4.1.5 Video de introducción

Video que contiene la narrativa del juego. Se dispone en orientación *landscape* y tiene un botón de “saltar” en la esquina superior derecha.

2.4.1.6 Ventana de derrota

La ventana de derrota se activa cuando el jugador pierde en el juego. Contiene los botones de “juego terminado” y “vida extra”, además de la puntuación efectuada en el juego recién pasado.



Figura 2.12: Ventana de derrota Astro Kitty. Fuente: Elaboración propia.

2.5 Arte

2.5.1 Estilo

El estilo artístico es “cartoon”. Los colores vivos son una constante en el juego, ya que entrega felicidad y además hace contrapeso a la idea de aterrizar forzosamente en un planeta desconocido.

2.5.2 Arte conceptual



Figura 2.13: Título del juego y logo. Fuente: Paperseed.



Figura 2.14: Astro Kitty viajando por el espacio en su nave. Fuente: Elaboración propia.



Figura 2.15: Astro Kitty despistado previo al choque con un asteroide. Fuente: Elaboración propia.

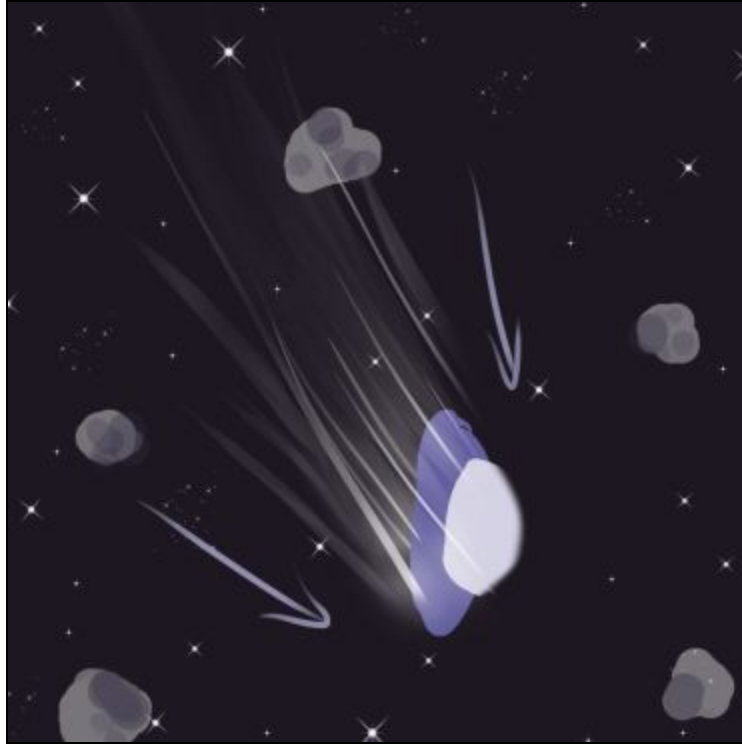


Figura 2.16: Astro Kitty se prepara para un aterrizaje forzado. Fuente: Elaboración propia.



Figura 2.17: Astro Kitty divisa algo en el horizonte. Fuente: Elaboración propia.



Figura 2.18: Una cueva con repuestos de nave. Fuente: Elaboración propia.

2.5.3 Sonido y música

Astro Kitty se sitúa en un contexto moderno, dado que lo referente al espacio es un símbolo de ello, en consecuencia la música del juego es electrónica con un “beat” marcado que genera cohesión con la mecánica principal del salto: llevar un ritmo constante al saltar, para perfeccionar el control del personaje.

La música se encuentra sin cortes en toda la escena 2 (ver figura 2.6) , pero su volumen aumenta en el nivel para acentuar la inmersión en el jugador.

Ideas generales: Aventura, desafío, loop.

Los sonidos referentes al juego son los siguientes:

- Click en botones (*sfx_button*)
- Recolección de repuestos de nave (*sfx_score*)
- Salto de Astro Kitty (*sfx_jump*)
- Superar récord de puntaje (*sfx_highscore*)
- Condición de derrota (*sfx_die*)

2.5.4 Animaciones

Las animaciones presentes en *Astro Kitty* son las siguientes:

Logo: Animación de rebote

Botón “vida extra”: Animación corazón palpitando

Personaje:

- Ocio: Cuando no salta el personaje
- Salto: Cuando se efectúa la mecánica de salto

2.5.5 Fuentes

El juego consta de dos fuentes para el juego.

Rocco: Usado en el título del juego



Figura 2.19: Fuente Rocco. Fuente: Dafont.

Bauhaus: Usado en el puntaje, puntaje más alto, *gameover* y "vida extra".



Figura 2.20: Fuente Bauhaus 93. Fuente: Dafont.

2.5.6 Sistema de partículas y efectos especiales

Paralaje: El efecto de mover diversos planos a velocidades distintas genera un efecto de profundidad de campo atractivo para el jugador.

Estela de repuesto de nave: El repuesto de la nave tendrá una estela que replica su trayectoria.

Sistema de partículas de repuesto de nave: El jugador al hacer contacto con un repuesto de nave, crea un efecto de partículas con estrellas para fortalecer la sensación de la obtención de puntaje.

2.6 Calendarización

15-02 : Inicio etapa producción

- 15-02/07-03: Núcleo del juego
- 08-03/28-03: Flujo de juego
- 29-03/15-04: Publicidad y corrección de errores.
- 16-04/08-05: Sonido, video y efectos especiales.

09-05/11-05: Publicación Google Play

Las etapas mencionadas se desarrollarán bajo una metodología ágil donde cada fecha corresponderá a un hito de desarrollo.

En el siguiente capítulo se procede a estructurar el trabajo para el desarrollo de Astro Kitty y deconstruir el documento de diseño para la asignación de tareas a los miembros del equipo de trabajo.

Capítulo 3: Etapa de producción de videojuego

3.1 Metodología de desarrollo

Para cumplir con la calendarización del documento de diseño fue necesario estructurar la producción con un marco de trabajo ágil, principalmente porque se necesitaba una mecánica principal pulida en la fase de publicación: cada iteración de los hitos descritos en el punto 2.6 necesitaban una fase de *testing* por parte del equipo de trabajo, como usuarios que podrían jugar el videojuego. Otro factor a considerar es que el equipo de trabajo no tiene un espacio físico determinado de trabajo, por lo tanto era necesario transparentar el avance del proyecto con un software de planificación.

El manifiesto ágil [20] tiene cuatro valoraciones importantes:

- El foco se debe poner más sobre las personas y las interacciones que sobre los procesos y herramientas.
- El software funcionando es más importante que la documentación extensiva.
- La colaboración con el cliente es un eje central para el producto final.
- El proceso debe responder a los cambios, en lugar de seguir un plan.

Teniendo estas consideraciones, se decide por la metodología de desarrollo Kanban⁷ para este trabajo, ya que en algunas situaciones el avance del proyecto se vuelve intangible e invisible, es por ello que el tablero Kanban ayuda a hacer visible la etapa de producción con los diversos colaboradores, que a su vez, tenían conocimientos básicos de esta metodología en trabajos anteriores.

3.1.1. Kanban y su caracterización en el equipo de trabajo

Según Atlassian [21], Kanban consta de cinco componentes principales:

Señales visuales: Son tarjetas con los componentes del desarrollo del proyecto, las que pueden ser historias de usuario, mecánicas, *sprites*⁸, animaciones, entre otros. En concreto,

⁷ Kanban viene de la palabra japonesa “señal visual”, es un sistema de tarjetas que permite orientar el flujo de trabajo.

⁸ Es una imagen usada para representar un ente gráficamente (o parte de él) y poder posicionarlo en el lugar deseado de una escena mayor.

Astro Kitty presentaba tarjetas en el área de programación, ilustración, musicalización y efectos sonoros y animación.

Columnas: Las columnas representan una actividad específica que en conjunto componen el *workflow*. Las tarjetas fluyen a través del *workflow* hasta su finalización. La caracterización de las columnas en el desarrollo del videojuego Astro Kitty es la siguiente:

- Por hacer (equivalente a *backlog*).
- WIP⁹.
- Urgente.
- Revisión.
- Completado.

Límites de WIP: Son el límite máximo de tarjetas que pueden estar en un momento dado. Por ejemplo si una columna con un límite de WIP de 4, implica que no puede tener más de 4 tarjetas en esa etapa del flujo de trabajo. Cuando la columna está “al máximo”, el equipo necesita mover las tarjetas hacia adelante antes de que las nuevas puedan llegar a esa etapa del flujo de trabajo. De esta forma se expone de una manera visual los cuellos de botella. Debido a que Paperseed no presenta experiencia previa en el desarrollo de videojuegos, no trabajo con esta característica de una manera rígida, sin embargo la calendarización otorga información clave para detectar cuellos de botella en el desarrollo.

Punto de compromiso: Es el momento en el que el equipo recoge los requerimientos del trabajo y los comienza a ejecutar. De acuerdo a la calendarización mencionada en el punto 2.6 existen 4 puntos de compromiso.

- 15-02: Núcleo del juego.
- 08-03: Flujo de juego.
- 29-03: Publicidad y corrección de errores.
- 16-04: Sonido, video y efectos especiales.

Punto de entrega: Es el final del flujo de trabajo de un tablero Kanban. Si bien para la mayoría de los equipos el punto de entrega es cuando el producto está en manos del cliente, que en este caso sería la publicación en Google Play. Paperseed define 4 puntos de entrega correspondiente a los hitos mencionados anteriormente:

- 07-03: Núcleo del juego.
- 28-03: Flujo de juego.
- 15-04: Publicidad y corrección de errores.
- 08-05: Sonido, video y efectos especiales.

⁹ *Work in progress.*

El tablero Kanban fue completado por el autor de esta memoria en conjunto con el modelador 3D, quien en éste trabajo, no tenía incidencia debido a que es un juego en dos dimensiones; y la comunicación del equipo era de al menos 2 reuniones semanales para evaluar los estados de avance del proyecto.

3.1.3 Software de planificación

Los tableros digitales permiten a los equipos que no comparten un espacio físico (como Paperseed) gestionar el flujo de trabajo de manera remota. Del abanico de posibilidades existente en el mercado, se decide el uso del *software* gratuito Trello para manejar la producción de Astro Kitty.

Trello es una herramienta gratuita que permite simular una pizarra con tarjetas, como un tablero Kanban, siendo extremadamente potente para entornos colaborativos. Es posible asignar fechas y miembros a las tarjetas, que además pueden ser arrastradas por el tablero simulando el *workflow*. Además el *software* es multi-plataforma permitiendo a los usuarios estar al tanto de las constantes actualizaciones del desarrollo. Por último, cada tarjeta puede tener componentes que especifiquen aún más las características del proceso en cuestión: *checklist* para tareas que se puedan desglosar en procedimientos pequeños; adjuntar archivos; y asignación de etiquetas para transparentar aún más la información son algunas de las cualidades que presenta la versión gratuita. La figura 3.1 es un tablero de Trello estándar que usa el modelo Kanban.

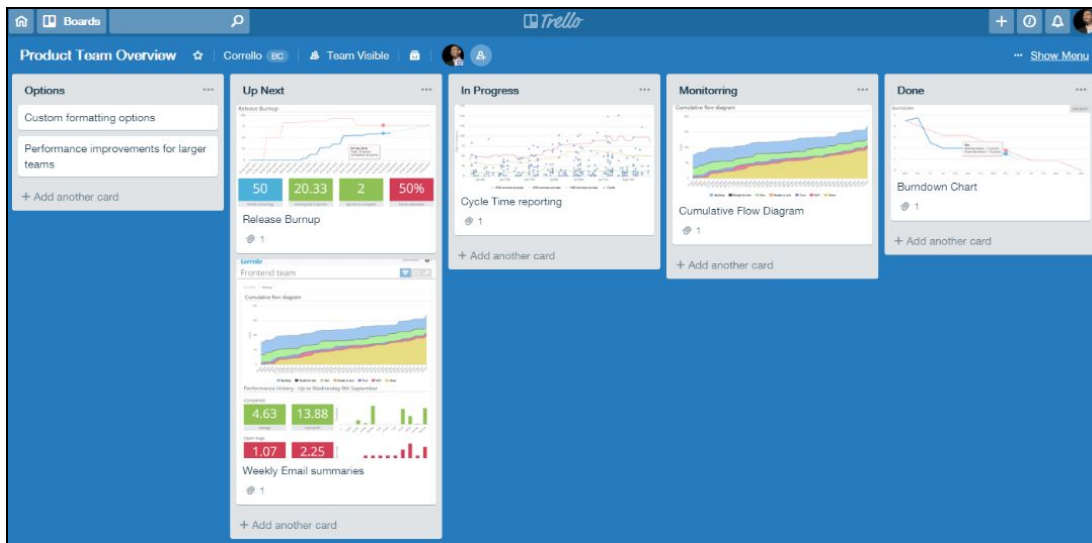


Figura 3.0: Tablero de Trello. Fuente: Trello.

3.2 Producción

El proceso iterativo de producción consta de las siguientes etapas:

Núcleo del juego: Componentes referentes a la jugabilidad de Astro Kitty. Es la iteración más extensa del proceso productivo.

Flujo de juego: Compuesto principalmente por el *GameManager* [19], *script* que maneja las reglas del juego.

Publicidad y corrección de errores: Elemento que gestiona la publicidad dentro del juego, posee una estructura similar al *GameManager*. Además se mejoran y corrigen elementos de la experiencia de juego.

Implementación sonora, video y efectos especiales: Etapa que concierne a la implementación de un *SoundManager* [19] y mejoramiento gráfico de la experiencia de juego.

Publicación: Configuración del sistema Google Play para subir los archivos correspondientes del videojuego Astro Kitty.

Antes de comenzar el desarrollo del proyecto es necesario configurar la escena y la estructura de archivos, para poder llevar de manera ordenada el trabajo.

En primer lugar, se debe definir el espacio de trabajo para dos dimensiones, de manera que se determinen configuraciones óptimas para el desarrollo posterior, como por ejemplo, la cámara ortográfica, que a diferencia de una con perspectiva, ésta posee una profundidad fija por lo que el aumento o reducción de *zoom* no tiene efectos.

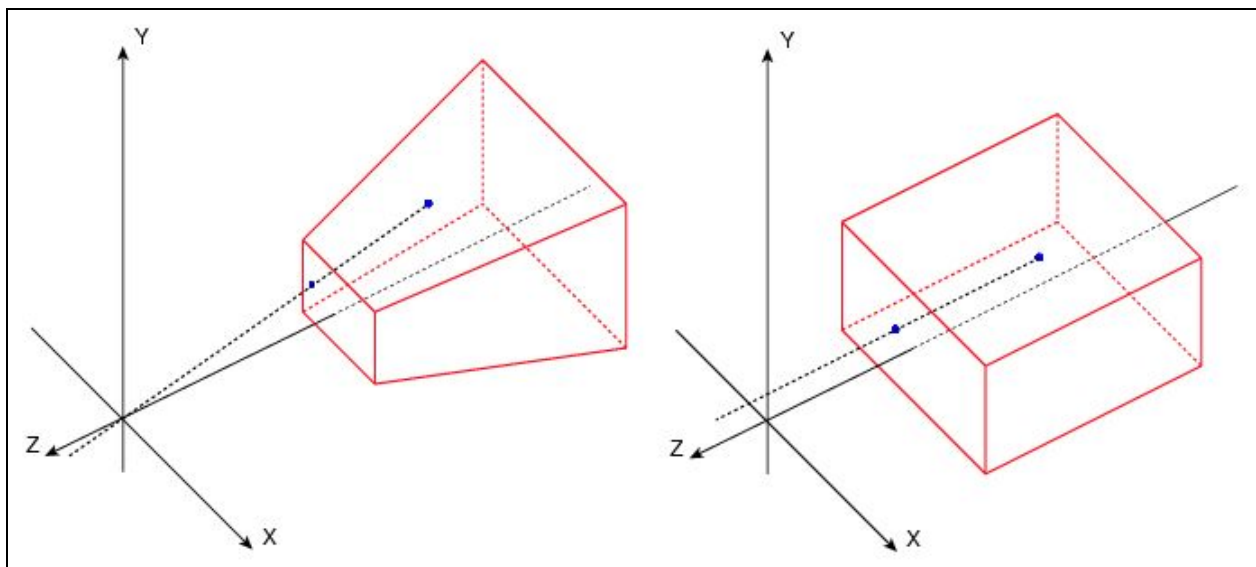


Figura 3.1: Cámara perspectiva y cámara ortográfica. Fuente: StackExchange.

La estructura de archivos del proyecto es la siguiente:

Animations: Contiene las animaciones del proyecto.

Fonts: Contiene las fuentes del proyecto.

Prefab: Módulo que contiene los *prefab*¹⁰ del juego.

Scenes: Carpeta con las escenas del proyecto.

Scripts: Contiene los archivos C# que dan funcionamiento a los *gameobjects*. Se crean subcategorías para una mayor organización.

Sound: Carpeta con los archivos de sonido y música del juego.

Sprites: Contenedor de los archivos gráficos de dos dimensiones. Se organiza en subcarpetas basándose en la profundidad de campo (ver figura 2.11), interfaz de usuario y el gato con sus animaciones.

UnityAds: Módulo de Unity para el manejo de publicidades en el juego. Se importa desde la Asset Store.

Videos: Contiene el video con la historia de Astro Kitty.

En general, y sobre todo para proyectos más grandes, es indispensable crear una taxonomía orientada a objetos para las estructuras de mayor nivel.

3.2.1 Núcleo del juego

Haciendo una recapitulación de los objetos plasmados en el GDD, ésta iteración se compone de los siguientes elementos:

Correspondiente al área artística está el desarrollo de la caverna, obstáculos, repuestos de nave y las animaciones del gato.

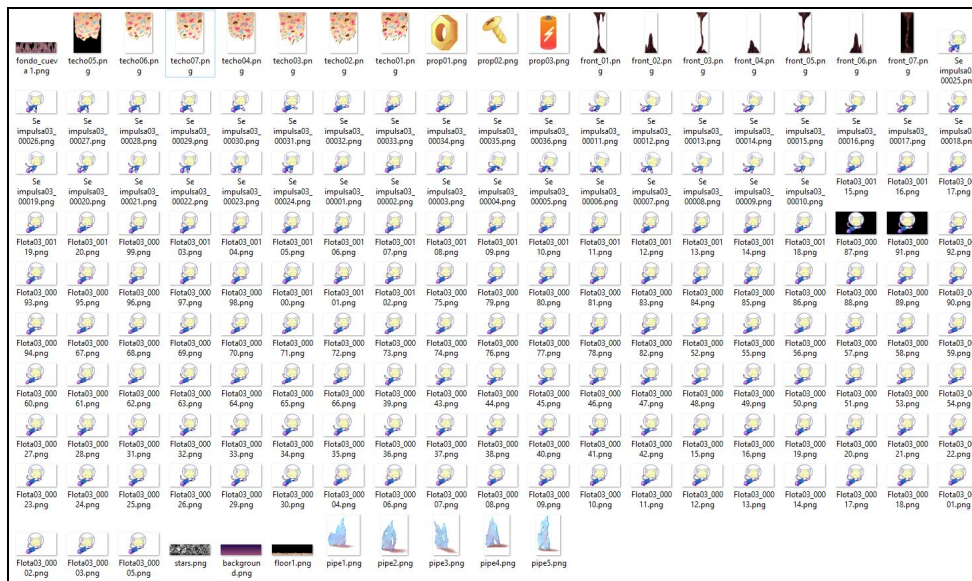


Figura 3.2: Elementos correspondientes al núcleo del juego. Fuente: Paperseed.

¹⁰ El sistema *prefab* de Unity permite crear, configurar y almacenar un *gameobject* completo con todos sus componentes, valores de propiedad y *gameobjects* secundarios como un activo reutilizable. El archivo prefabricado actúa como una plantilla desde la cual se pueden crear nuevas instancias en la escena.

Respectivo al desarrollo de las mecánicas y su implementación:

Movimiento de la caverna: Se utiliza el componente *Transform*¹¹ para generar el movimiento, éste implementado en la función *Update*¹². La particularidad, es que el objeto al salir del rango de visión de la cámara se vuelve a relocalizar a la derecha dando la sensación de una caverna infinita. Puntualmente, el techo de la caverna al ser relocalizado, varía en altura.

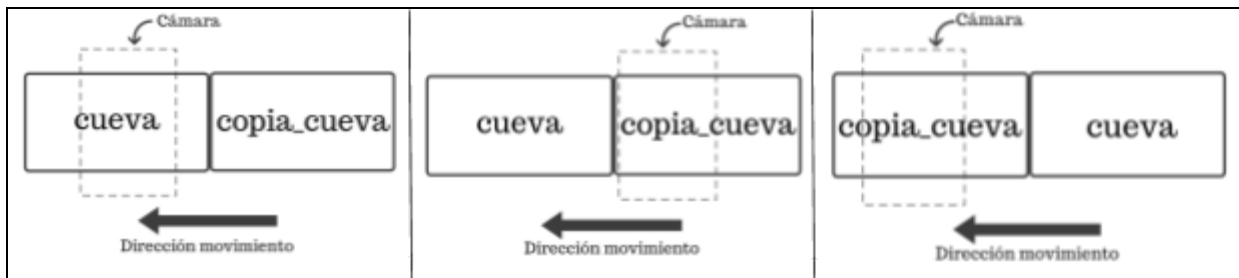


Figura 3.3: Movimiento de la caverna y relocalización. Fuente: Elaboración propia.

Este *script* se implementa en los objetos que son parte de la caverna y sólo corresponde asignar una velocidad distinta a cada uno de ellos para crear el efecto de paralaje¹³.

Creación y movimiento de los repuestos de nave: Se utiliza el componente *Transform* para ejercer el movimiento del objeto, donde el eje X se orienta con una constante; y el eje Y presenta una función seno para seguir un comportamiento pendular.

Para la creación de dichos objetos a un ritmo constante se implementa la función *Instantiate*¹⁴ con un temporizador que permite controlar la creación de objetos.

Es necesario manejar el “colector de basura”, por lo que todos los repuestos de nave son destruidos finalizada una sesión de juego.

Creación y movimiento de los obstáculos: El movimiento de éste objeto está condicionado por el espacio donde el personaje pueda pasar sin problemas. Para ello se utiliza la función *Raycast*¹⁵ como lo grafica la figura 3.6, en donde además del movimiento lateral, el obstáculo presenta un movimiento vertical que se detiene cuando el rayo colisiona con el techo de la caverna. De esa manera se definen distintos largos de *Raycast* para los obstáculos, logrando las dificultades de juego mencionadas en la figura 2.3.

¹¹ Componente en Unity que permite alterar la posición, rotación y escalado de los objetos.

¹² Función que hereda de “MonoBehaviour” y que se “llama” cada *frame*.

¹³ Paralaje es una técnica en computación gráfica donde las imágenes de fondo pasan más lentamente por la cámara que las imágenes del primer plano, creando una ilusión de profundidad en un escenario de dos dimensiones.

¹⁴ *Instantiate*: Función en Unity encargada de crear clones de objetos

¹⁵ *Raycast*: Rayo invisible con una posición, dirección y tamaño definido que permite detectar colisiones.

En una primera instancia se implementa la misma solución para manejar la creación de obstáculos, que es reemplazada en procesos iterativos posteriores, por un patrón de optimización llamado *Object Pooler*.

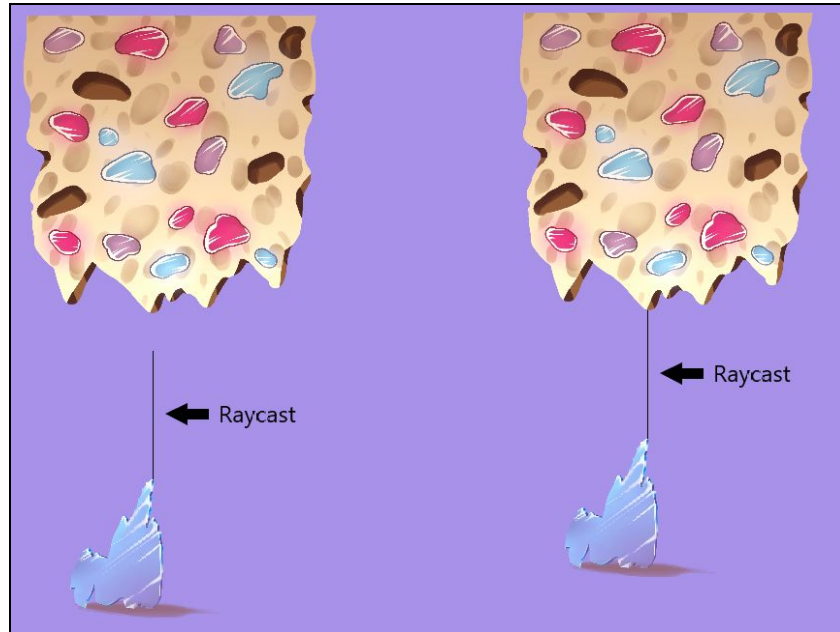


Figura 3.4: Movimiento vertical de obstáculos. Fuente: Elaboración propia.

Control del juego: El control del juego gatilla tanto la acción de salto, que es manejado por el módulo de físicas de Unity con el componente *Rigidbody2D*, como el sistema de animaciones del juego, que es gobernado por el componente *Animator*.

Manejo de colisiones del gato: *Collider2D* permite detectar las colisiones de los objetos que tengan dicho componente, permitiendo gatillar acciones con la función *OnTriggerEnter2D*. Así el personaje al colisionar con el *Midground* de la caverna se activa la condición de derrota.

3.2.2 Flujo de juego

Ésta iteración contiene desarrollo del menú, los estados del juego y el puntaje, y es donde el área artística desarrolla menos componentes, permitiéndole avanzar en el video introductorio que se implementará en el último punto de compromiso.

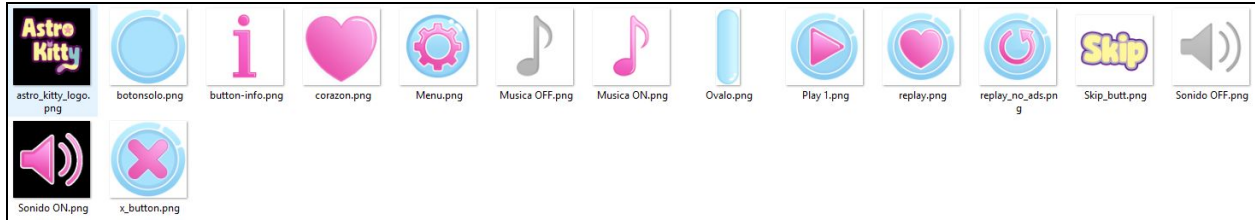


Figura 3.5: Componentes de la UI. Fuente: Paperseed.

Correspondiente al área de programación e implementación se desarrollan los siguientes elementos:

GameManager: Es un *Singleton*¹⁶ que maneja las reglas y estados del juego, que de acuerdo al punto 2.4.1 son los siguientes:

Start: Ventana correspondiente al menú principal.

Game over: Ventana de derrota.

Countdown: Ventana de cuenta regresiva.

Play: Ventana de juego.

Estos 4 estados de juego son expresiones de la UI en forma de *GameObject*, que se activan y desactivan de acuerdo a las reglas del juego. Además, este *script* controla el puntaje del jugador y guarda en memoria el más alto.

Otra característica fundamental de esta clase es el funcionamiento del núcleo de juego en el momento adecuado: tanto el movimiento de la caverna, como la instanciación de los repuestos de nave y los obstáculos deben “activarse” cuando el jugador esté en la ventana de juego.

3.2.3 Publicidad y corrección de errores

Sistema de publicidad: Unity provee un sistema completo de servicios integrados para crear juegos de manera colaborativa, incrementar productividad, monetización de juegos y gestionar la audiencia. Para este hito, se utilizó el SDK de Unity para monetizar juegos móviles llamado *Unity Ads*. El sistema se importa fácilmente desde la *Asset Store*¹⁷ e inicialmente debe ser configurado desde el panel de control de los servicios de Unity para definir qué tipo de publicidad tendrá el juego (ver figura 3.8).

¹⁶ Singleton: Patrón de diseño de software que restringe la creación de instancias de una clase a una instancia única.

¹⁷ Asset Store: Tienda para desarrolladores en Unity.

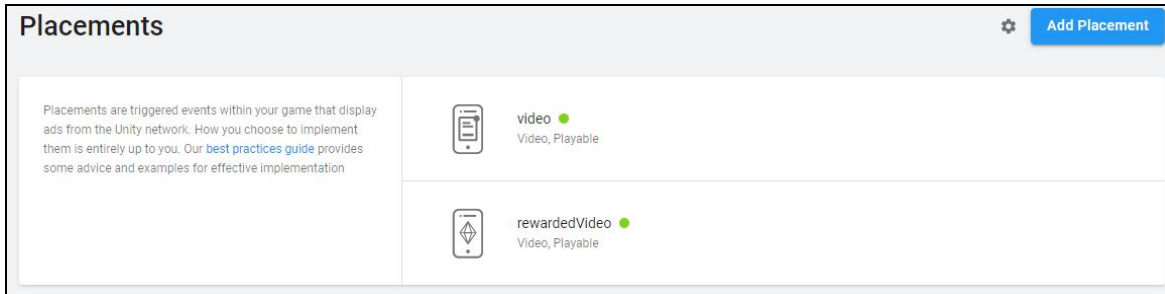


Figura 3.6: Tipos de publicidad definidos para Astro Kitty. Fuente: Elaboración propia.

El sistema de publicidad está controlado por un *AdsManager*, que también es un *Singleton*. Esto permite “llamar” a sus métodos desde el *GameManager* en dos momentos específicamente:

1. Cuando el usuario luego de 3 sesiones de juego, pasa de la ventana de derrota al menú principal.
2. Cuando el usuario desde la ventana de derrota apreta el botón de vida extra.

En esta etapa del proyecto el equipo de trabajo detecta una primera divergencia de la etapa de producción con el documento de diseño: si el usuario tiene la posibilidad de pedir una vida extra cada vez que muere, entonces el desafío de obtener el puntaje más alto está condicionado por la motivación de ver “video recompensas”, llegando a la fase de aburrimiento del producto mucho más rápido, debido a que la destreza pasa a ser un segundo plano. Por lo tanto, se re-configura el menú principal para que cada vez que se pida una vida extra, la próxima ventana de derrota no contenga el botón para volver a reclamarlo. La solución implementada fue crear un estado más a los señalados en el punto 3.2.2 llamado *Game Over Ads*, que sigue siendo la ventana de derrota pero con el botón con “video recompensa”.

Además, en este hito se implementa un mejoramiento de soluciones de la experiencia de juego como lo es el *Object Pooler*, y un nuevo *Raycast* para los obstáculos de la caverna.

Object Pooler: Viene a reemplazar la acción de instanciar y destruir los obstáculos de la caverna. Para ello se construye una “piscina” con objetos inactivos (obstáculos de caverna) de tamaño definido que se van activando en la medida que el juego los vaya demandando. Cuando estos objetos salen del rango de la cámara se vuelven a ubicar en la piscina (inactivos) esperando ser usados nuevamente. De esta manera se reusan los objetos y no se asigna memoria basura.

Raycast 2.0: La mejora correspondiente al *Raycast* tiene su motivación en que la distancia generada por éste elemento no era estática y depende de la morfología del techo de la caverna. En algunos casos era tan angosto el espacio entre los objetos que era imposible

pasar con el gato. La solución mejorada orienta el *Raycast* de manera horizontal pero se define la posición de manera manual en el editor de escenas de Unity. De esta manera se normaliza el espacio entre los objetos y la experiencia de juego no es frustrante.

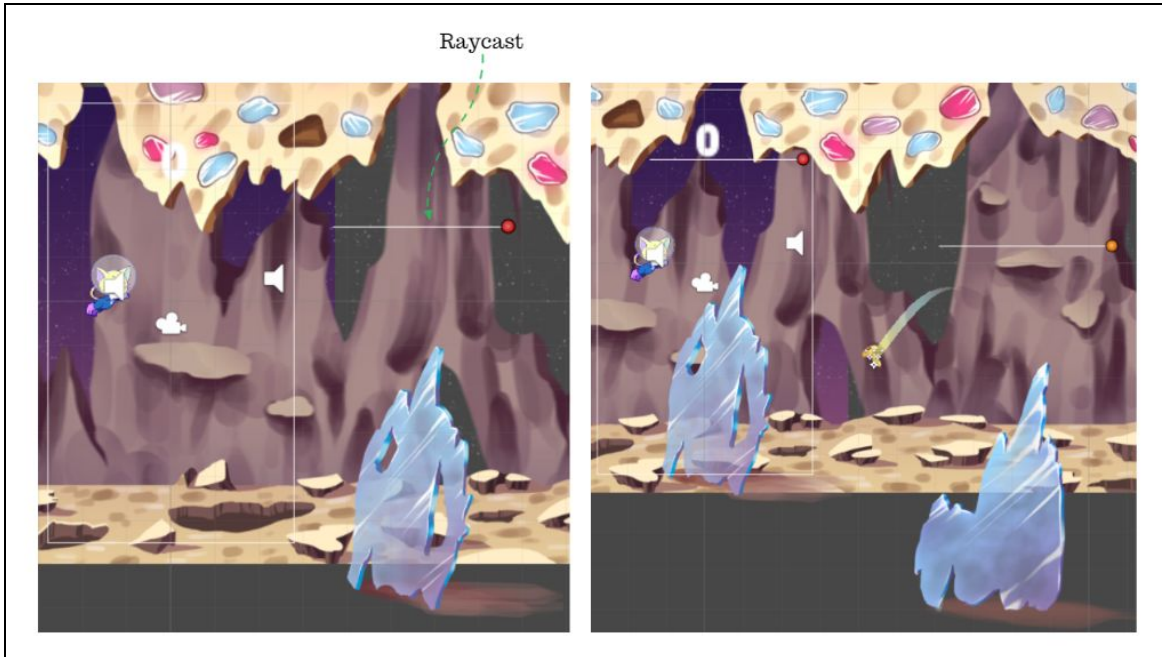


Figura 3.7: Solución mejorada al Raycast. Fuente: Elaboración propia

3.2.4 Sonido, video y efectos especiales.

AudioManager: *Singleton* encargado de la gestión de los efectos sonoros y música del juego, en donde métodos como control de volumen o silenciar sonidos son desarrollados en esta sección.



Figura 3.8: AudioManager en el Inspector de Unity. Fuente: Elaboración propia.

Video: Se incorpora al proyecto el video de introducción en una escena distinta a la del juego. Éste video se basa en el arte conceptual del punto 2.5.2.

Trail Renderer: Componente que le adhiere una estela a los objetos que la posean. Se implementa este elemento en los repuestos de nave para hacerlos más atractivos a la vista.

Sistema de Partículas: Unity posee un completo sistema de partículas que permite entregarle una mayor retroalimentación al jugador de los repuestos de nave.

Suavizado de movimiento: Cuando una sesión de juego termina, la caverna y el gato son reubicados en su posición inicial. La primera iteración de esta mecánica era una simple “teletransportación” de los objetos, pero se decide una interpolación lineal de la posición actual con la inicial, creando un suavizado de movimiento y denotando la nueva sesión de juego.

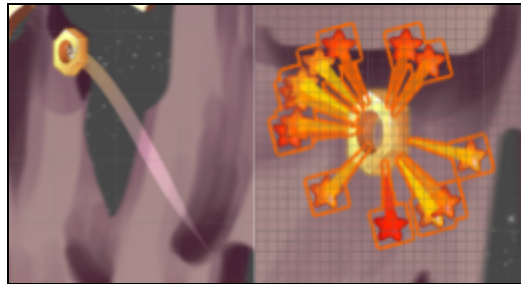


Figura 3.9: Efectos especiales en repuestos de nave. Fuente: Elaboración propia.

3.2.5 Problemas y optimización

El equipo de trabajo presentó problemas principalmente en la zona de juego, comenzando con la construcción de la caverna, debido a que el techo fue diseñado sin una anchura suficiente para ejecutar un corrimiento vertical, limitando la experiencia de juego en la navegación del gato.

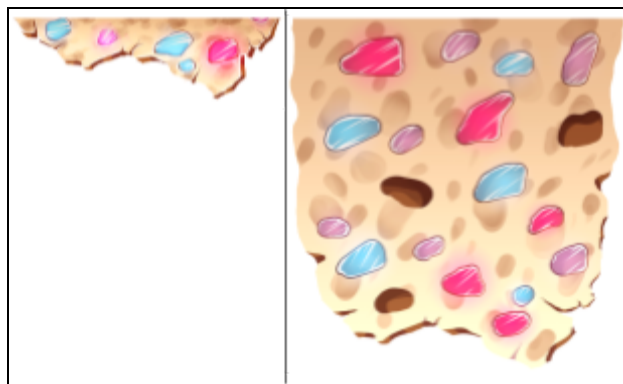


Figura 3.10: Evolución techo de la caverna. Fuente: Elaboración propia.

El siguiente problema a resolver fue la separación de los componentes del fondo de la cueva a medida que el juego avanza. Para explicar porqué sucede ese comportamiento es necesario entender que un *frame* es un ciclo de una serie de eventos que se ejecutan con un

orden específico en Unity y que es variable en cuanto al tiempo (a diferencia de una película). El instante de tiempo del frame anterior se obtiene gracias a la propiedad *deltatime* de la clase *Time*, que además permite mover objetos usando el componente *Transform*. Por lo tanto, la distancia avanzada por frame es también variable y el punto crítico se sitúa al momento de relocalizar los objetos, como lo ilustra la figura 3.5.

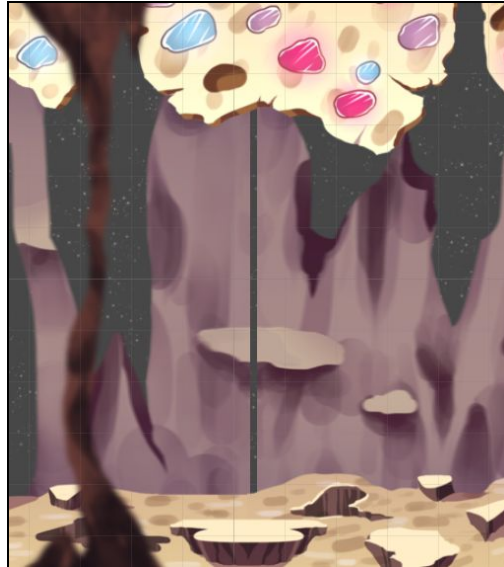


Figura 3.11: Separación de elementos de la cueva al relocalizar. Fuente: Elaboración propia.

La solución al problema solapando ligeramente los componentes de manera que ese diferencial no se perciba a simple vista.

Además del *Object Pooler* y la nueva implementación del *Raycast* en el punto 3.2.3, se optimiza la configuración de los componentes básicos del juego. Unity permite importar archivos en gran calidad y reducirlos en el *Inspector*¹⁸. De esa manera todos los elementos fueron optimizados a más del 50% de su resolución original (que en algunos casos era 8k). Por otra parte, los clips de sonido también fueron modificados, basándose en la frecuencia de uso y tamaño de los archivos [22].

Con respecto a la composición de los sonidos, se rehizo el sonido de la muerte del gato debido a que sugería la destrucción de una nave espacial y el tema principal no presentaba un ritmo marcado ni sonidos artificiales.

¹⁸ *Inspector*: Muestra información detallada sobre el *GameObject* seleccionado actualmente, incluidos todos los componentes adjuntos y sus propiedades, y le permite modificar la funcionalidad de *GameObjects* en su escena.

3.2.6 Testing

Los ciclos de prueba fueron cerrados a un grupo cerrado de personas y entregaron retroalimentación diversa:

- La dificultad del juego era muy alta y creaba espacios imposibles de pasar: Se optimiza con la optimización del punto 3.2.3.
- La cuenta regresiva antes de comenzar el juego era muy larga: Se redujo de 3 segundos a 2 segundos.
- La publicidad aparecía muy seguido: Se cambió la frecuencia con que aparecen la publicidad intersticial de 2 a 3 sesiones de juego.

Capítulo 4: Etapa de publicación de videojuego

4.1 Android

El proceso de publicación de aplicaciones Android comienza con una serie de configuraciones en Unity ubicadas en la sección “*Player Settings*”, siendo la de mayor importancia la creación del *keystore* [23], ya que sin él, es imposible realizar el proceso de publicación.

Otra consideración importante fue que desde este año Google implementó un nuevo formato de publicación para incluir el código de compilación y los recursos de una aplicación llamado *Android App Bundle*. Este paquete de archivos permite tener APKs más optimizadas dependiendo de las características técnicas de cada equipo, seleccionando solo los componentes que necesita.



Figura 4.0: Lógica de instalación de componentes de Android App Bundle. Fuente: Google.

La versión usada de Unity para este proyecto es compatible con esta nueva implementación de Google, según lo afirma este artículo [24] y solo era necesario habilitar la funcionalidad en la ventana “*Build Settings*” y las arquitecturas ARMv7, ARM64 y x86.

El panel de control de Android consta de 5 pestañas esenciales para la correcta publicación de contenido en la tienda Google Play:

Versiones de la App: Gestiona los archivos .aab (Android App Bundle), el control de versiones de la aplicación, alpha cerrado y beta abiertas a los usuarios de Google Play.

Para efectos del proyecto se realizó una beta cerrada fuera de este sistema y se realiza un lanzamiento oficial del producto.

Ficha de Play Store: Contiene información del producto tales como el título, la descripción, recursos gráficos, categoría, contacto y política de privacidad.



Figura 4.1: Recursos gráficos usados en la publicación de Astro Kitty. Fuente: Elaboración propia.

Clasificación de contenido: Google Play contiene un sistema de clasificación de contenido de aplicaciones y juegos que es calificado mediante un cuestionario de la International Age Rating Coalition (IARC[25]).

Astro Kitty se categoriza para todas las edades según el cuestionario.

Precios y distribución: Maneja el precio y la distribución de la aplicación en los 149 países que Google Play tiene presencia. Además dicha selección permite seleccionar proveedores de telefonía e internet móvil en algunos casos. También en este apartado es necesario

explicitar si la aplicación contiene anuncios y aceptar las leyes de exportación de los Estados Unidos.

Para efectos de este proyecto se define Astro Kitty como una aplicación gratuita con presencia en todos los países y con contenido de anuncios publicitarios.

4.2 Herramientas de analítica

Una vez ya publicado el juego, tanto Google Play como Unity presentan servicios que permiten seguir el rastro de la aplicación en diversas categorías. A continuación se detallan las características principales de estos sistemas de información que dan soporte a toda la gestión de las operaciones futuras.

4.2.1 Google Play Console

Panel de estadísticas: Controla la información relacionada de los usuarios, segmentación por dispositivos, calificaciones en la tienda, control de calidad, control de pre-registro e ingresos por compras dentro de la aplicación.

Android Vitals: Maneja la información de las fallas, los ANR¹⁹ de la aplicación y tamaño de la aplicación.

Herramientas de desarrollo: Permite el uso compartido de aplicaciones a nivel interno y la incorporación de servicios de juego de Google Play.

Adquisición de usuarios: Lleva un informe de adquisición de los usuarios considerando los alcances orgánicos, campañas de Google, canales monitoreados, búsqueda en Google y referencias de terceros. Además la creación de campañas en Google Ads, promociones y sugerencias de optimización de la aplicación.

Informes financieros: Muestra un gráfico de los ingresos generados a lo largo del tiempo.

Comentarios de usuarios: Lleva un registro de las calificaciones, comentarios y análisis de reseñas mediante palabras clave.

¹⁹ ANR: *Android Not Responding*.

4.2.2 Panel de control de Unity

Vista General: Presenta los principales indicadores económicos del videojuego que corresponden a las ganancias totales, ARPDAU, eCPM²⁰, Impresiones²¹, Promedio DAU²², Retención²³.

Reportes: Sección con gráficos económicos de la aplicación, ganancias publicitarias, IAP y promociones.

Monetización: Guarda relación con la configuración e instalación de los métodos de monetización de Unity Ads. Se hace uso de esta característica en el punto 3.2.3.

Optimización: Sección que permite realizar experimentos controlados que expone a diferentes grupos de jugadores a variaciones de algún aspecto del juego. Además permite configuraciones remotas y cambiar variables de la aplicación sin realizar actualizaciones.

Analítica: Permite acceder a estadísticas de la aplicación y dar seguimiento al comportamiento que tienen los usuarios dentro del juego configurando previamente eventos en el motor de videojuego. De esta manera se obtiene información sensible para realizar procesos de actualización y mejorar indicadores clave.

4.2.3 Retroalimentación de usuarios

Las fuentes de recopilación de usuarios tienen dos orígenes: las reseñas efectuadas en la plataforma Google Play luego de publicar el videojuego y una encuesta realizada en el evento realizado en Valparaíso el 18 de Mayo de 2019 llamado “Fan Valpo”.

4.2.3.1 Reseñas

El juego tiene 37 calificaciones con un promedio de calificación de 4.892. 22 de estas calificaciones tienen comentarios que se detallan en la siguiente tabla:

Nº	Fecha	Dispositivo	Comentario
1	8/5/2019 a la(s) 16:44	Galaxy J5 Prime	“Muy buen juego!!! 100% recomendable”
2	8/5/2019 a la(s) 16:45	Galaxy S7	“Refrescante y entretenido!!! 7v7”
3	8/5/2019 a la(s) 16:54	Galaxy S7	“Tremendo!!! muy buena

²⁰ eCPM: Effective Cost Per Thousand Impressions es una métrica utilizada para medir el rendimiento de monetización de anuncios de un *publisher* de aplicaciones, y es calculada mediante la ecuación (total ganancias/total impressions) x 1000.

²¹ Impresiones: Es una instancia publicitaria en el videojuego.

²² DAU: Daily Active Users.

²³ Retención: Porcentaje de usuarios que continúan interactuando con la aplicación a lo largo del tiempo.

			animacion, musica agradable, excelente jugabilidad y diseño, perfecto para momentos de ocio, diversión asegurada.”
4	8/5/2019 a la(s) 17:50	POCO F1	“Astro Kitty sobresale en su acabado artístico y musical, no obstante el concepto y jugabilidad se asimilan bastante al de otros juegos dentro del género, siendo repetitivo después de unos minutos. A nivel técnico, la fluidez y animaciones cumplen, pero estas últimas podrían trabajarse mejor en ciertas ocasiones como, por ejemplo, cuando el gatito "muere", pues tener el sprite inicial flotando en el mismo espacio rompe demasiado la inmersión.”
5	8/5/2019 a la(s) 18:07	Galaxy S7	“This first version is quite good, I look forward to the next update to have a pleasant surprise for us. Keep that way Paperseed, congratz”
6	8/5/2019 a la(s) 21:21	Galaxy J5	“Excelente juego. Muy entretenido y fácil de jugar”
7	8/5/2019 a la(s) 22:25	Galaxy J7 Prime	“Personaje adorable! ♡ y entretenido juego ♡”
8	9/5/2019 a la(s) 15:08	Moto G(4)	“Me encanto la musica y el diseño gráfico. Los avisos pueden saltarse en solo pocos segundos!! . Esperemos ver que añadan mas cosas”
9	11/5/2019 a la(s) 15:01	Galaxy J7	“Esta entretenido solo le falta implementar más cosas pero para ser un juego reciente esta genial nwn”
10	11/5/2019 a la(s) 15:44	LG X max	“Excelente juego muy buena musica”
11	12/5/2019 a la(s) 1:09	Moto G (5th Gen)	“De mecánica sencilla e intuitivo, la

			música es bastante agradable”
12	12/5/2019 a la(s) 14:52	Galaxy S8+	“Ligero, incluso me trae buenos recuerdos notándose la inspiración a flappy, buenas animaciones y son gatos, solo le faltan trajes e iría 10/10 🐱🐱 Si lo tuyo es volver a remontar flappy, mejor descargar esta nueva joya en bruto”
13	12/5/2019 a la(s) 19:30	Galaxy J7 Prime	“Conocí el juego hoy en un evento, solo puedo decir que muy entretenido y con hartito cariño”
14	12/5/2019 a la(s) 20:41	Galaxy J7 Prime	“Excelente juego, muy adictivo y fácil de jugar, aunque yo le agregaría mas gatitos”
15	13/5/2019 a la(s) 11:37	Galaxy S7	“Juego muy adictivo y bello, añadiría opciones de customización a modo de premio ♡ por superar ciertos puntajes”
16	13/5/2019 a la(s) 13:00	Galaxy S4	“Me encanto el diseño la dinámica y la música de este jueguito, uno no se aburre y quiere aumentar su record jajaja sigan asi!!!!”
17	15/5/2019 a la(s) 10:38	Y550-L03	“Mis sobrinos aman este juego. es muy bonito felicidades”
18	16/5/2019 a la(s) 17:55	Galaxy S7	“It's cute and fun!!”
19	17/5/2019 a la(s) 0:50	Galaxy Grand Prime Plus	“Muy bueno recomendado”
20	17/5/2019 a la(s) 2:17	Moto E (4) Plus	“ES MUY BONITO 🍷🍷🍷🍷🍷”
21	2/6/2019 a la(s) 22:22	P20 lite	“Serían 5 pero me estresa que solo con recoger las pilas y tal ganas puntos y no pasando los "palos" pero de igual manera me encanta < 3”

Tabla 4.0: Comentarios de Astro Kitty en Tienda Google Play. Fuente: Elaboración Propia.

Además el sistema tiene un análisis de reseñas que captura los comentarios con palabras claves como música, jugabilidad, personaje, entre otros y evalúa el efecto de esa calificación en el promedio general.

espacio	espacio (Español)	3.000 ★		1		
minutos	minutos (Español)	3.000 ★		1		
jugabilidad	jugabilidad (Español)	4.000 ★		2		
personaje	personaje (Español)	5.000 ★		1		
musica	musica (Español)	5.000 ★		4		

Figura 4.2: Sistema de análisis de reseñas de Astro Kitty. Fuente: Google Play Console.

4.2.3.2 Encuesta

La encuesta se realizó a 12 personas con el fin de evaluar aspectos específicos de la versión lanzada en Play Store. Las preguntas son las siguientes:

Datos personales:

1. Edad
2. Sexo
3. ¿Juegas videojuegos móviles?
4. ¿Cuántas horas al día?

Preguntas sobre el juego:

1. La mecánica del juego, ¿es fácil de entender?.
2. La interfaz, ¿es fácil de entender?
3. ¿Le pareció el juego divertido?
4. ¿Jugaría este juego en sus momentos de ocio?
5. ¿Siente invasiva la publicidad dentro del juego?

A través de esta encuesta se extrajo la siguiente información:

1. 58.3% de los encuestados es mayor de edad.
2. Sólo un 16.7% de los encuestados son niños.
3. 50% de los encuestados son mujeres.
4. Un 66.7% de los encuestados juegan videojuegos móviles.
5. Un 72.7% de los encuestados juega menos de 2 horas al día videojuegos.
6. Un 91.7% considera que la mecánica y la interfaz del juego fácil de entender.

7. Un 83.3% de los encuestados considera el juego divertido.
8. Un 91.7% jugaría Astro Kitty en sus momentos de ocio.
9. Con respecto a la publicidad, un 41.7% considera que la publicidad no es invasiva, un 33.3% considera que tal vez puede ser invasiva y solo un 25% si la considera invasiva.

4.2.4 Versiones posteriores

Las versiones posteriores de Astro Kitty incluyen cambios pequeños en la interfaz y algunas correcciones en el escenario para minimizar el problema .

- Se solapan aún más los componentes de la caverna para evitar el problema mencionado en el punto 3.2.5.
- Se cambia el texto “Try Again” por “Extra Life!” y se añade una animación de latido al corazón para impulsar al jugador a presionar el botón.
- Se añade el botón “salir del juego” porque se reporta que algunos usuarios que poseen el panel de navegación en la pantalla no podían acceder a él desde el juego.

Además, se planea como equipo de trabajo una actualización mayor para mejorar la retención de los jugadores y añadir componentes a la jugabilidad.

Capítulo 5: Conclusiones

5.1 Mercado

La instalación de los juegos hiper-casuales como un nuevo género es sin duda, la consideración más importante del mercado, teniendo 2 hitos que el autor de esta memoria estima como determinantes en su expansión. El primero, es la aparición de Flappy Bird, el cual masifica esta nueva experiencia de entretenimiento interactivo y es caso de estudio de empresas emergentes que ven un “océano azul[26]” en las plataformas digitales. El segundo, es la potente inversión de Goldman Sachs en la empresa Voodoo quienes se posicionan como los líderes del mercado.

Por un lado, los valores de producción de estos juegos extremadamente bajos, pueden realizarse en pocas semanas, por otra parte su atractivo para la masividad y una monetización ultra agresiva de anuncios, aseguran los ingresos de un proyecto exitoso: entre 10 a 100 millones de dólares un éxito hiper-casual podría generar en un tiempo relativamente corto[27]. Así, muchas empresas se subieron al carro de este nuevo género, fortaleciendo algunas estructuras en la cadena de valor, para una mayor adquisición de usuarios, logrando convertirse en una de las tendencias del 2018.

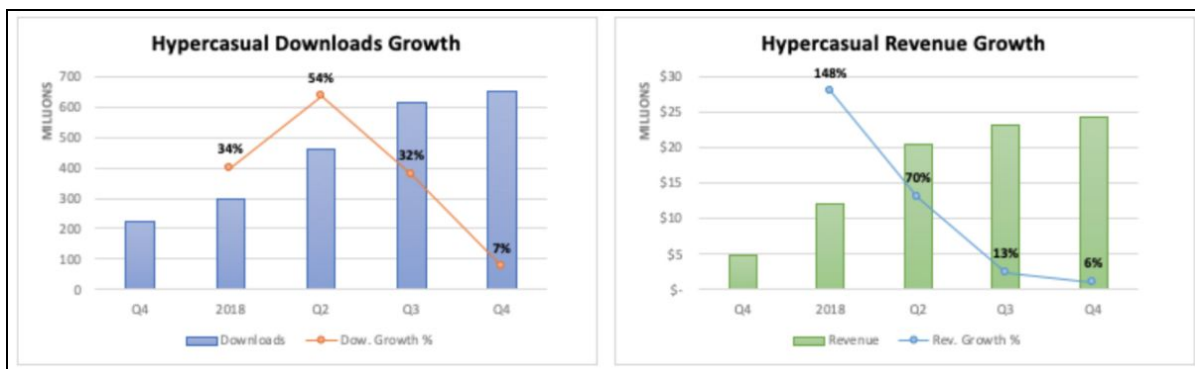


Figura 5.0: Evolución del mercado hiper-casual el año 2018. Fuente: Deconstructor of fun.

Pero toda esta agilidad en el desarrollo y monetización, permite que la competencia se apresure y la diferenciación del producto se vuelve más difícil e importante, la retención de

los jugadores va a la baja, el *payback*²⁴ aumenta y el ROI²⁵ disminuye: el mercado ha madurado[28].

Esta desaceleración en el mercado produce que la entrada a la competencia como el poder de permanencia ahora son más difíciles, y tanto Voodoo como Ketchapp, quienes dominaban el mercado a principios de 2018, han perdido gradualmente su cuota de participación, esto debido a que los principales productores de juegos están experimentando presión al encontrar títulos nuevos y frescos que puedan replicar sus éxitos pasados. Además nuevas empresas entran en competencia, como Applovin, quienes realizan un brillante movimiento estratégico para la optimización de ingresos publicitarios y crecimiento en la adquisición de usuarios, abriendo un brazo de *publishers* llamado Lion Studios[28]. Es un hecho, la aparición de nuevas empresas publicadoras de juegos hiper-casuales donde ellos solo gestionan una porción de la operación: administrar el movimiento y tener rentabilidades positivas en el ecosistema.

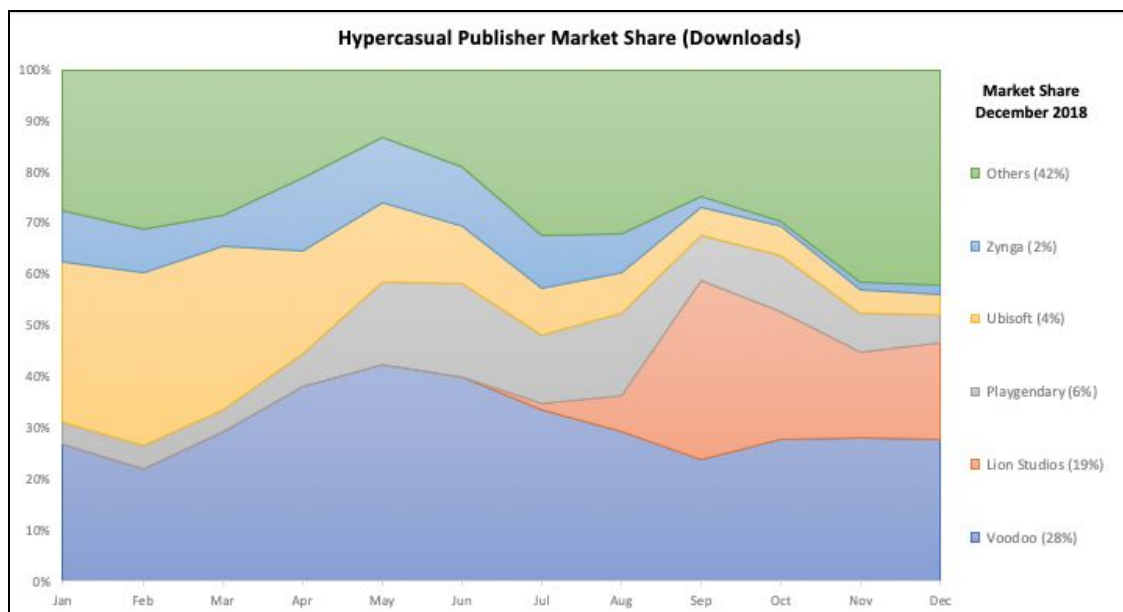


Figura 5.1: Cuota del mercado hiper-casual año 2018. Fuente: Deconstructor of fun.

Esta competitividad del mercado, obliga a las empresas desarrolladoras de juegos hiper-casuales crear capas de *meta-game* más atractivas para mejorar la retención de los usuarios, desarrollar productos con una mejor calidad gráfica y de jugabilidad y, explorar las IAP como nueva forma de monetización.

²⁴ Payback: el tiempo que toma recuperar el costo de una inversión.

²⁵ ROI: *Return On Investment* es un ratio que permite medir lo invertido en un negocio y el beneficio obtenido de dicha inversión.

En esa línea, según el estudio de IronSource llamado “*The truth about Hyper-casual*” [28], quienes analizaron los datos de 2.5 mil millones de usuarios en su red durante 2 años, descubren que 660 millones de usuarios juegan hiper-casual, 520 millones juegan tanto IAP como hiper-casual, pero curiosamente 101 millones del grupo mencionado anteriormente jugaron primero un juego hiper-casual. Esto quiere decir, que este 20% que jugó un juego hiper-casual en un principio, luego se traslada a IAP, preparándose en cierta medida para juegos más robustos.

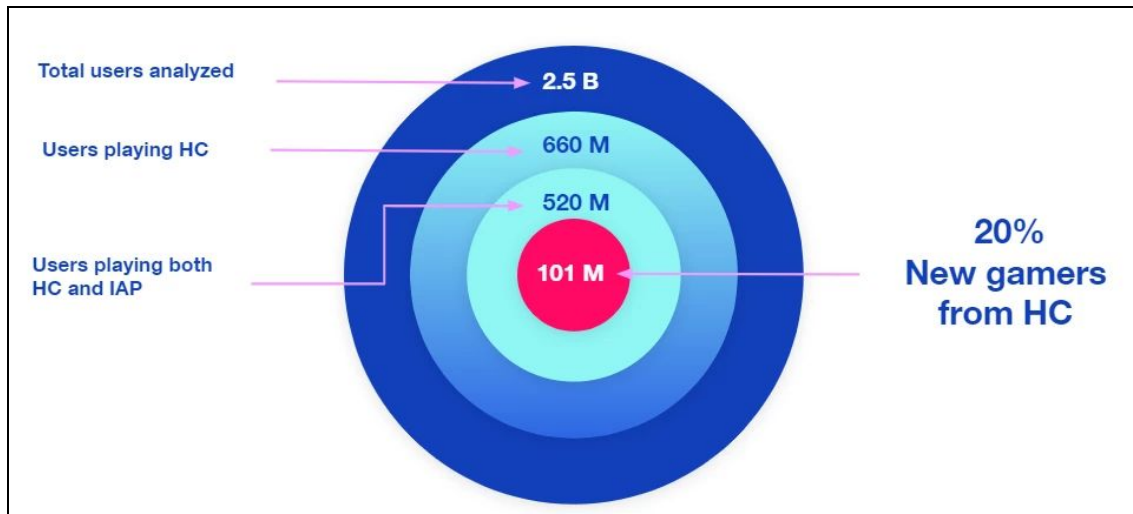


Figura 5.2: Muestra de usuarios por categoría de juego. Fuente: IronSource.

Este nuevo inventario de personas que juegan hiper-casual ve 4.8 anuncios de video promedio, siendo 2 veces más que las otras categorías; instalan en promedio 10 veces más aplicaciones en general (por 1000 usuarios activos diarios) e instalan 5 veces más aplicaciones IAP que los usuarios que juegan otras categorías [28]. Esto brinda una gran oportunidad para que los juegos IAP adquieran nuevos usuarios. Depende de ellos asegurar que realicen compras dentro del juego.

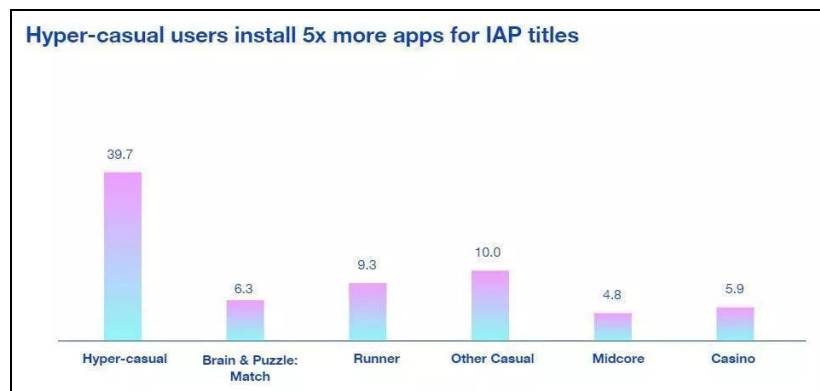


Figura 5.3: Caracterización del usuario hiper-casual. Fuente: IronSource.

Este enorme crecimiento de instalaciones trae consigo un aumento en la publicidad emitida. La figura 5.4 revela que el año 2018 hubo aumento de las impresiones emitidas en juegos hiper-casuales y un decrecimiento de las IAP. En muchos sentidos, el inventario hiper-casual representa un segmento atractivo para los anunciantes ya que son vitrinas con mayor concurrencia de público, que además se ve fortalecido por la promoción cruzada.

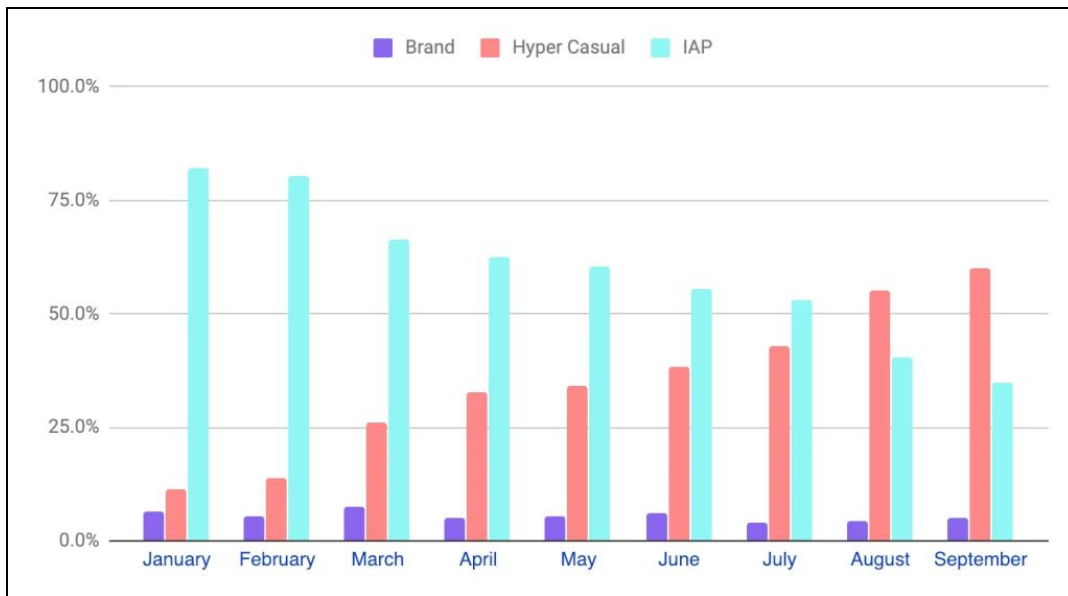


Figura 5.4: Cantidad de impresiones emitidas por segmento. Fuente: IronSource.

A pesar del gran crecimiento de los juegos hiper-casuales, son insostenibles como género independiente. En algún momento se debe generar dinero a medida que los usuarios se mueven de un juego a otro y en el mundo hiper-casual esto no sucede lo suficiente. La única manera de sostener este mercado es que tanto los anunciantes de IAP como los de marcas, busquen clientes en el segmento hiper-casual, o este tipo de juegos exploren otros métodos de monetización y se conviertan en experiencias híbridas para suplir ese déficit.

Dentro de estas experiencias híbridas de monetización, algunos ya comenzaron a realizarlas como el juego *Tomb of the Mask* y *Kick the Buddy* de Playgendary, quienes mejoraron sus ganancias vía IAP en un 70% (+5.5m) los últimos dos trimestres del año 2018 [27] utilizando el sistema de compra de objetos y suscripciones semanales.

Considerando los avances tecnológicos y la próxima llegada de 5G con la actualización permanente de las capacidades de los dispositivos móviles, el director de operaciones de IronSource, Tal Shoham, estima que habrá un corrimiento hacia el desarrollo de juegos *Hardcore*, ya que se presentan condiciones más propicias para su jugabilidad: el tiempo de instalación de un juego de 800 MB es una barrera de entrada para la tecnología actual,

además de prescindir de conectividad toda la experiencia de juego. Ya existen algunos casos del género Battle Royale como Fortnite, PUBG y Garena Free Fire, que se logran posicionar como las de mayor recaudación en las listas de las tiendas digitales. Esta no es una predicción hiper-casual, pero sí marca un precedente de cómo el ecosistema va evolucionando a experiencias más complejas.

Por último, debido al rápido crecimiento de los videojuegos hiper-casuales en el mercado, las empresas de publicidad han creado plataformas de inteligencia de negocios robustas para optimizar indicadores económicos, granularizar los tipos de usuario y ofrecer experiencias automatizadas que agilizan el trabajo. Además se espera un aumento en el tipo de publicidades tales como anuncios 360°, realidad aumentada, realidad virtual, anuncios jugables, entre otros.

5.2 Pre-producción

En primer lugar, es necesario recalcar que el documento de diseño no tiene una estructura, ni formato rígido, y debe ser adecuado al alcance del proyecto, por lo tanto es beneficioso tener un pensamiento modular que permita definir esa estructura con facilidad. Además, la incorporación de imágenes, videos, sonidos y mapas mentales es crucial para llevar un documento de diseño cohesionado en los miembros del equipo.

Una de las características del rol del diseñador de videojuego es lograr entender cómo las mecánicas y reglas se combinan para obtener un producto divertido. Una aproximación formal para entender los videojuegos es el *framework* MDA[18] (*Mechanics, Dynamics and Aesthetics*), que transcribe los componentes como lo ilustra la figura 5.5.

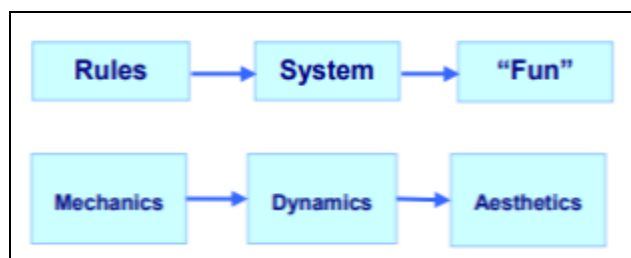


Figura 5.5: Componentes de un juego y su contraparte en el framework. Fuente: MDA Framework.

Las mecánicas son los componentes del juego a nivel de la representación de datos y algoritmos. Las dinámicas corresponden a la interacción de las mecánicas en el tiempo de ejecución; y la estética que describe las respuestas emocionales (deseadas) evocadas en el jugador. Esta representación permite esclarecer que al momento de crear un juego, tanto el diseñador de videojuegos como el jugador, tienen perspectivas distintas de lo que perciben

primero: desde la perspectiva del diseñador, las mecánicas creadas dan origen a las dinámicas que se convierten en una experiencia estética en particular, mientras que desde la perspectiva del jugador, la estética entona lo que son las dinámicas observables y eventualmente las mecánicas operables (ver figura 5.6).

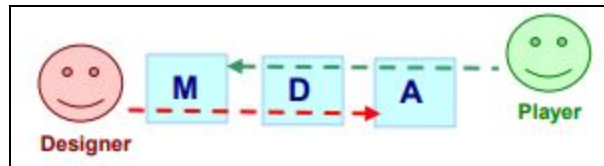


Figura 5.6: Perspectivas del diseñador y jugador en un videojuego. Fuente: MDA framework.

Se puede apreciar en Astro Kitty cómo la mecánica del movimiento ejecutada en distintas velocidades genera la dinámica de paralaje o profundidad de campo, además de la sensación del movimiento del personaje. Si sumamos las mecánicas de colisión y de puntaje se genera una estética de desafío, donde cada sesión de juego tiene como objetivo superar el puntaje anterior. Es notorio que los puntos de estrés del jugador están en los momentos que el personaje intenta pasar los obstáculos.

El *framework* MDA aborda fases iterativas de desarrollo de mecánicas para experimentar y descubrir las sensaciones, impacto y emociones que generan en los jugadores, controlado por un sistema de retroalimentación que permite evaluar resultados tangibles. Esta estructura encaja perfectamente con las metodologías ágiles como en la implementada en Astro Kitty, sin embargo, dado que la experiencia de juego es reducida, hubiera sido mejor realizar 2 o más fases iterativas en el núcleo de juego para afinar la composición del escenario y éste ajustarlo con la mecánica principal de salto. Además el sistema de retroalimentación que contempla el *framework*, que en este caso es el sistema de puntaje, podría haber sido un sistema híbrido que contenga los repuestos de nave y la distancia avanzada en la sesión de juego, permitiendo medir con mayor certeza la experiencia de un jugador.

Con respecto a los anuncios publicitarios, a final de cuentas, es una decisión de negocio que varía dependiendo del funcionamiento del juego en el ecosistema, por lo tanto, si bien es parte de la etapa de diseño, su calibración se desarrolla en iteraciones posteriores. Sin embargo, es necesario tener la consideración que el vídeo recompensa debe verse como una característica del producto y debe tratarse como tal.

Dada que la retención de los jugadores en Astro Kitty es baja, principalmente porque la jugabilidad del juego es acotada, es prudente pensar en nuevas capas de jugabilidad y *meta-game* para mejorar ese indicador. Se propone, a grandes rasgos, utilizar el puntaje

(repuestos de nave obtenidos) como una llave para desbloquear nuevos aspectos del gato, videos con narrativa que vayan acompañados de nuevas experiencias en jugabilidad: escenario nuevo con obstáculos y/o enemigos que afecten en el puntaje final de la sesión. La implementación de plataformas sociales como Facebook o Google Play también son anexos que fortalecen el juego como una experiencia comunitaria.

Del análisis anterior se extrae que las capas de *meta-game* fortalecen nuevas experiencias estéticas, sobre todo en los juegos hiper-casuales que carecen de una jugabilidad robusta: nuevos aspectos del gato fortalecen la idea del juego como una experiencia para expresarse; los videos con narrativa van de la mano con la visión del juego como una historia; y las plataformas sociales como un juego comunitario. Del lado de la jugabilidad, niveles nuevos, fortalecen la estética de descubrimiento: el juego como un territorio no explorado.

Por otro lado, si se analiza el mercado Voodoo, éste ha mantenido un diseño visual ultra-minimalista privilegiando una mecánica exitosa. Esta estructura hegemónica de hacer juegos se contrarresta con el diseño visual de Astro Kitty, que si bien hoy en la actualidad esto puede verse como una amenaza, los cambios en el mercado pueden forzar a buscar experiencias innovadoras en el aspecto visual.

Para finalizar esta sección, la documentación de mecánicas de otros juegos en una biblioteca de *assets*, permite la agilización del diseño y una mayor especificidad de lo que se pretende producir: al diseñar una mecánica en específico, por ejemplo la de salto, es posible encontrarse con una cantidad variada de tipos de salto como lo menciona Polygon[30], y para ello el qué, cómo y porqué deben estar explicitados en el documento de diseño.

5.3 Producción

El dinamismo en el mercado de los juegos hiper-casuales, obliga a las empresas de desarrollo manejar metodologías ágiles para tener prototipos de jugabilidad pulidos. Paperseed, en la práctica, plantea una estructura híbrida que combina las tarjetas visuales, con iteraciones productivas (planteadas en la calendarización del documento de diseño), un *backlog* que orienta las tareas futuras y una metodología de lanzamiento como lo maneja *Scrum*. *Scrumban* o *Kanplan* son aproximaciones a la metodología implementada y actualmente existen herramientas de *software* que personalizan el trabajo de esta forma. La implementación del documento de diseño no tuvo mayores contratiempos, sin embargo detectar algunos comportamientos no deseados tiende a ser complejo, porque la solución

puede ser abordada por el diseño gráfico de los componentes, la disposición de los objetos en la escena o la programación de los comportamientos.

Si bien se intenta detallar minuciosamente los elementos e interacciones del juego, es inevitable (quizás por la falta de experiencia) encontrarse con divergencias en la etapa de desarrollo. Por ejemplo, y luego de analizar la caverna, es posible que la morfología de ésta (en algunas situaciones) no se ajuste a la experiencia de juego deseada. Este tipo de situaciones es necesario documentarlas para tener conocimiento futuro en el caso de experiencias similares a futuro.

Por otra parte y siendo el alcance del videojuego pequeño, no es necesario un trabajo de optimización tan exhaustivo: Por una parte Unity ofrece regular la calidad de los *sprites*, video y sonido; y la técnica de *Object Pooling* no logra realizar una mejora tangible, principalmente porque la frecuencia de instanciación de los objetos (3 segundos) no ocasiona que la llamada de la función Destroy asigne memoria basura.

En este contexto se realiza un experimento con la “creación de objetos” cada 0.1 segundos, para ambos casos presentados en 3.2.2 y se comprueba que el *Object Pooling* es más óptimo que la solución tradicional. Se deduce que es un patrón de diseño para frecuencias altas como instanciación de balas, hordas de enemigos, etc.

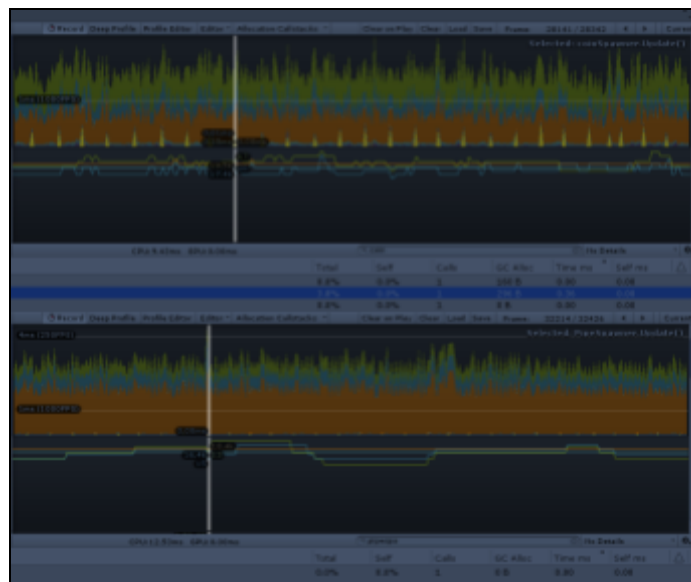


Figura 5.7: Comparativa del profiler en ambos métodos de instanciación de objetos. Fuente: Elaboración propia.

Con respecto al sistema de publicidad, éste se implementa sin saber cómo funcionan las mecánicas de monetización en el ecosistema. Uno de los grandes factores para obtener ganancias en un videojuego en base a publicidad es el porcentaje de solicitudes de anuncios que las redes publicitarias poseen y se mide con un ratio llamado *fill rate*. Cada vez que un usuario solicita mediante un video recompensa o intersticial y no se ejecuta, son pérdidas potenciales para la empresa. Esto se puede solucionar con 2 implementaciones:

1. La implementación de varias redes publicitarias (UnityAds, IronSource, AdMob, etc) que actúen como plan B,C o D en el caso que la primera no logre entregar el anuncio. Esto implica la integración de varios SDK's en el proyecto y su aumento de tamaño, además de presentar informes fragmentados y dispares. La tienda de Unity ofrece varias herramientas que operan bajo esta solución, sin embargo no es la óptima.
2. La configuración de *Ad Mediation* que viene a ser una solución optimizada de la primera y donde se integra un sólo SDK que permite configurar, administrar y optimizar múltiples redes publicitarias.

5.4 Publicación

En términos generales el proceso de publicación presentó un único problema: La primera versión del juego estuvo publicada sin problemas durante 2 días, pero luego fue eliminada por no cumplir con las políticas de privacidad de Google Play. Y es que en Mayo de 2019 se obliga a todas las aplicaciones poseer una política de privacidad que proporciona a los usuarios el cómo se utilizan sus datos confidenciales y de sus dispositivos, agregando a la descripción de la aplicación una URL válida con las políticas de la empresa [30]. Por lo tanto, se realiza una primera actualización de la aplicación con los cambios demandados por Google Play el 10 de Mayo de 2019.

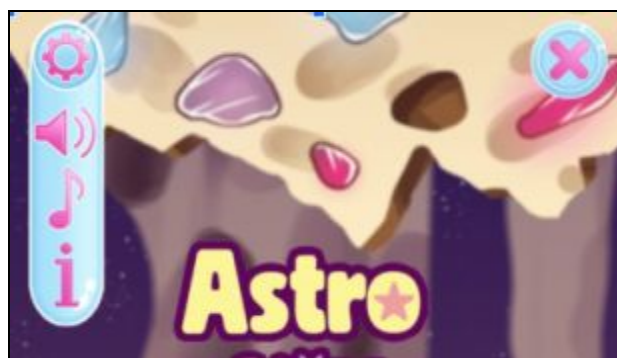


Figura 5.8: Botón de información con la política de privacidad de Paperseed. Fuente: Elaboración propia.

De esto se deduce que el sistema de Google Play es dinámico puesto que se rige por la normativa vigente de cada país. Un caso emblemático de este tipo de situaciones es la actualización de políticas correspondiente a los botines de las cajas de recompensas. Esta

capa de *meta-game* es una forma de monetización que permite al usuario obtener de manera aleatoria objetos que mejoran la experiencia de juego, que si bien, tiene sus primeras apariciones en el año 2004, se hace más visible con el juego Overwatch de Blizzard el año 2010. Recién el año 2017 en Bélgica se comienza a legislar sobre esta forma de obtener ingresos [31], aludiendo a que el azar usado en esta mecánica es equivalente a apostar en un casino, y que en mentes no desarrolladas, como en niños, puede generar adicción [32]. De esta forma, la política de Android recibe una nueva actualización: “Las aplicaciones que ofrecen mecanismos para recibir artículos virtuales aleatorios de una compra (es decir, “cajas de botín”) deben revelar claramente las probabilidades de recibir esos artículos antes de la compra”, sumado a una nueva política de familias que entra en rigor el 1 de Septiembre de 2019. En conclusión, es necesario estar al tanto de las normativas vigentes y las actualizaciones futuras de las plataformas de distribución de aplicaciones.

Por otro lado, las herramientas de analítica proveen una visión clara del negocio y cómo el videojuego se desenvuelve en el ecosistema. La siguiente figura entrega la información de la cantidad de usuarios que instalaron Astro Kitty y su retención al día 1.

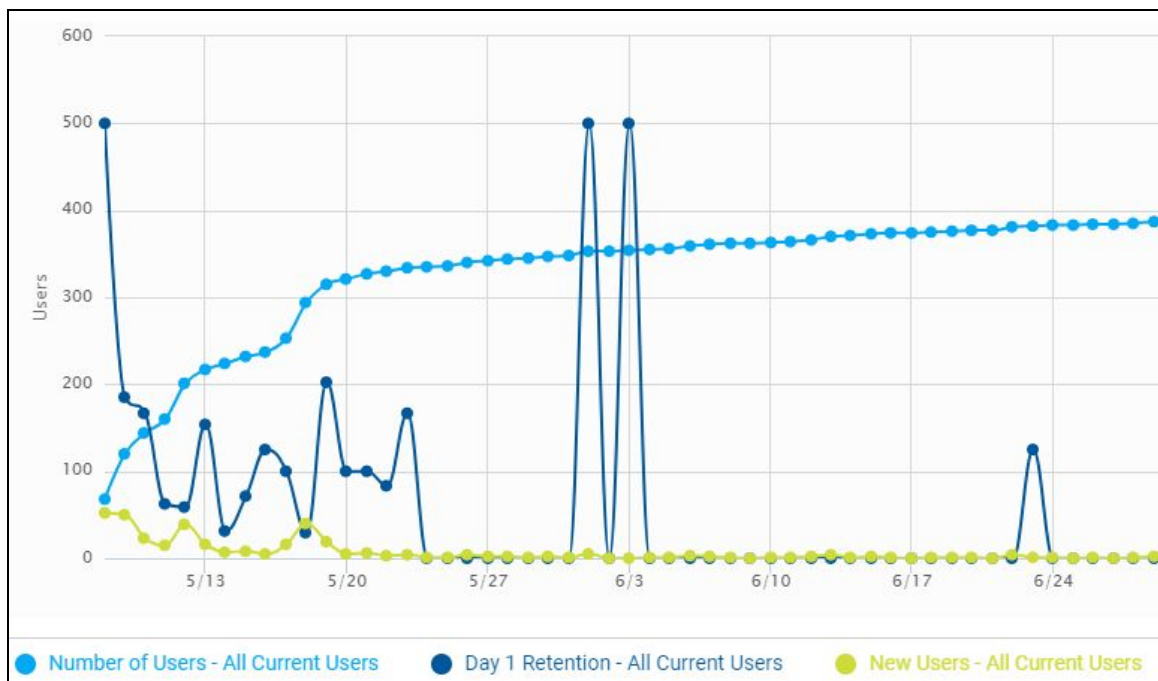


Figura 5.9: Cantidad de Usuarios y Retención D1 de Astro Kitty. Fuente: Unity Analytics.

Se puede observar que la curva celeste presenta 2 pendientes de crecimiento, correspondientes a las presentaciones del videojuego en ferias temáticas en Valparaíso: Kazoku Fest y Fan Valpo. Sin embargo, como el proyecto no contemplaba un plan de

adquisición de usuarios, principalmente por los costos asociados, los indicadores fueron a la baja y se puede observar que ya desde el día 27 de Mayo, se aprecia una ínfima participación de nuevos usuarios. Aún así, se logra un total de 398 usuarios adquiridos de manera orgánica.

Es necesario destacar la funcionalidad de Unity Analytics para estudiar el comportamiento de los jugadores en el *gameplay*. Se concluye, que en el caso de su incorporación, éste debería implementarse en la mecánica del puntaje, creando una segmentación de jugadores que pasen ciertos umbrales de puntuación predefinidos (así, se tendría una métrica de la cantidad de jugadores que logran superar los 20, 40 o 60 puntos, por ejemplo); además, la recopilación de las colisiones por parte del jugador con la caverna, en especial con los obstáculos de cristal, puede entregar información de qué segmento del nivel tiende a ser más difícil. De esta manera, se plantea una etapa de post-producción con métricas que permiten abordar las futuras actualizaciones con mayor certeza.

En lo referente a la encuesta realizada, es necesario ver cómo estos últimos años la percepción de la publicidad ha cambiado. Del estudio de López P. (2016)[19], se extrae que un 90% de los encuestados sostiene que la publicidad es un impedimento para volver a jugar el juego, mientras que en esta encuesta sólo un 25% considera la publicidad invasiva. Este cambio de apreciación se atribuye principalmente a que el medio en general se movió en esa línea: desde páginas web, redes sociales y videos online contienen publicidad como formas de monetización. Sin embargo, los videojuegos tienen el componente interactivo que permite convertir los anuncios en vidas extras, monedas, aspectos de personaje, entre otras cosas. Incluso ya se habla de ballenas publicitarias, quienes son jugadores que aprovechan cada publicidad que les ofrece el videojuego para mejorar su experiencia de juego.

5.5 Proyecciones y trabajo futuro

Actualmente la empresa sigue en operaciones y planea desarrollar 4 prototipos de videojuegos hiper-casuales a final de año para así fortalecer su portafolio y apoyarse con entidades gubernamentales que fomentan productos y servicios chilenos en el extranjero (PROChile).

Además se está evaluando la posibilidad de postular a fondos concursables, principalmente CORFO, para la creación de un paquete de juegos hiper-casuales que luego será presentado a *publishers* de este género para intentar lograr una relación contractual con ellos, como lo tiene Inner Child, empresa chilena que desarrolla videojuegos a Voodoo.

Esto va a contracorriente con los últimos casos apoyados por CORFO conocidos por el autor de esta memoria, que se caracterizan por proyectos ambiciosos de 2 a 3 años que no han permitido a dichas empresas levantar capital propio. De esta manera, se desarrollan prototipos de 2 a 3 semanas y se maximizan las oportunidades de éxito.

Por último, el equipo de trabajo planea implementar Scrum como nueva metodología para futuros proyectos, esto apoyado con un software de gestión orientado a videojuegos y agilidad llamado *HacknPlan*.

5.6 Reflexiones personales

El autor de esta memoria cree que se cumplen los objetivos específicos correspondientes a cada etapa del proceso de desarrollo de un videojuego, a pesar de llevar un trabajo en función de tareas y sin cumplir horario laboral normal. Sin embargo, esto conlleva a que la creación de una cultura organizacional no se desarrolle con la misma velocidad que en un caso ideal de trabajo. Aún así, Astro Kitty fue un proyecto sencillo que permitió agrupar conocimientos técnicos de diversa índole y convertirlo en un estudio de videojuegos independiente.

Del capítulo 1 se rescata principalmente la cantidad de datos y especificidad en las necesidades de los jugadores (motivado por que es un mercado de 122.000 millones de dólares), para así tomar decisiones correctas de negocio, como lo hizo Goldman Sachs en su tiempo. Dichos estudios demográficos y psicográficos que respaldan decisiones de desarrollo en el extranjero, deberían tomarse en cuenta al momento de apoyar proyectos chilenos mediante fondos concursables para maximizar oportunidades de éxito.

El capítulo 2 fue fundamental para enmarcar el trabajo, ya que proyectos anteriores donde no hay existencia de un documento de diseño eran ampliados sin horizonte y termina desgastando al equipo en tareas que perdían el foco del videojuego.

Cabe destacar que en estudios más grandes la existencia de un documento de diseño técnico, que es una bajada operativa de cómo abordar los requerimientos del proyecto, en este caso no se realiza principalmente porque el diseñador del juego y el programador son el autor de este trabajo.

Dentro del abanico de soluciones para el requerimiento del plano jugable, una malla procedural del techo y el suelo hubiera entregado un rango impredecible de espacios de juego, haciendo un juego menos monótono y podría esto haber mejorado la retención de

los jugadores. Sin embargo se optó por una solución más simple, principalmente porque era el primer proyecto del equipo.

Con respecto al capítulo 3 es necesario destacar que soluciones como la mencionada anteriormente se pueden encontrar en la tienda de *Assets* de Unity, permitiendo agilizar el trabajo, pero es necesario tomar resguardos debido a que la integración con los demás elementos del juego puede ocasionar comportamientos no deseados.

El proyecto en general no presentó “cuellos de botella”, principalmente porque los hitos eran holgados. Pero en condiciones normales de trabajo, la programación los hubiera presentado, por lo que al momento de finalizar el proyecto, el equipo decide buscar otro programador para maximizar el rendimiento del área artística, creando 2 juegos en paralelo. De igual manera, la reutilización de componentes es fundamental: tanto el *GameManager*, *AudioManager*, *AdsManager* son elementos recurrentes en cualquier juego hiper-casual, permitiendo acotar los tiempos de los proyectos. Eventualmente con el tiempo, la empresa pretende tener una librería con elementos de *meta-game* que serán reusados dependiendo del proyecto en cuestión.

El capítulo 4 se extrae que el proceso de post-producción, específicamente la monetización y la adquisición de usuarios deben ser ejecutados por ingenieros comerciales y/o industriales para llevar un control correcto de los procesos financieros en el ecosistema mercantil. Esta podría considerarse como una vía alternativa de crecimiento de la empresa *Paperseed*, pero supone grandes barreras de entrada (sueldos extras y campañas adquisición de usuarios elevada). Es por ello que actualmente las grandes empresas que dominan el mercado tienen un poder de negociación que les favorece: trabajan con estudios pequeños pero solo pagan por los proyectos que superen ciertos indicadores claves que garanticen la viralización del producto.

Estas prácticas poco éticas no son las únicas en el mundo de los videojuegos: desde hace unos años que los desarrolladores de videojuegos reclaman por mejores condiciones laborales, quienes han llegado a trabajar 100 horas semanales para terminar títulos que le entregan grandes arcas a dichas empresas. La solución más viable a esta desigualdad en la correlación de fuerzas es la sindicalización. Bernie Sanders, senador Estadounidense manifiesta lo mismo: “La industria de los videojuegos hizo 43 billones de dólares en ganancias el año pasado. Los trabajadores responsables por esas ganancias merecen negociar colectivamente como parte de un sindicato”.

Pasando al lado educativo, es necesario mencionar auge de iniciativas público/privadas en torno al pensamiento computacional y la programación en etapas más tempranas, donde la

educación STEAM es una fuerte expresión de ello y donde el desarrollo de videojuegos y los juegos hiper-casuales encajan perfectamente, con herramientas de programación gráfica como lo son Scratch²⁶ o Construct²⁷. Es labor de instituciones como la Universidad Técnica Federico Santa María, levantar proyectos que fomenten el razonamiento basado en evidencia, el pensamiento creativo y la apreciación del proceso de investigación científica en liceos y colegios.

Por último, manifestar el apoyo del Departamento de Informática en crear instancias para desarrollar conocimiento en áreas poco comunes como lo es, el desarrollo de videojuegos y tecnologías emergentes de realidad aumentada y virtual (con los ramos de “Proceso de creación de videojuegos” y “Taller de creación de videojuegos”), además de prestar soporte para la participación de eventos relacionados, permite al estudiante tener una visión competente para afrontar el mundo laboral en esta área.

²⁶ <https://scratch.mit.edu/>

²⁷ <https://www.scirra.com/>

Referencias

[1] *Number of apps in the iTunes app store since 2008, Statista.*

<https://www.statista.com/statistics/268251/number-of-apps-in-the-itunes-app-store-since-2008/>

[2] *The ascendance of hypercasual, PocketGamer.*

<https://www.pocketgamer.biz/comment-and-opinion/65256/the-ascendance-of-hypercasual/>

<https://www.pocketgamer.biz/comment-and-opinion/65390/hyper-casual-part-two-what-defines-hyper-casual/>

<https://www.pocketgamer.biz/comment-and-opinion/65618/the-ascendance-of-hyper-casual-the-opportunity/>

[3] *1 billion angry birds downloads, Rovio.*

<http://www.rovio.com/news/1-billion-angry-birds-downloads>

[4] *Why is Candy Crush Saga so popular?, The Guardian.*

<https://www.theguardian.com/technology/2014/mar/26/candy-crush-saga-king-why-popular>

[5] *Rethinking the MMO, Gamasutra.*

https://www.gamasutra.com/view/feature/1583/rethinking_the_mmo.php?page=3

[6] *Tower defense, Wikipedia.*

https://es.wikipedia.org/wiki/Tower_defense

[7] *Apptopia: Helix Jump led all mobile games in Q4 2018 with 25.6 million daily players, Venturebeat.*

<https://venturebeat.com/2019/01/17/apptopia-helix-jump-led-all-mobile-games-in-q4-2018-with-25-6-million-daily-players/>

[8] *Top 10 Game Mechanics for Hyper Casual Games, Mobile Free To Play.*

<https://mobilefreetoplay.com/top-10-game-mechanics-for-hyper-casual-games/>

[9] *Newzoo's Global Games Market Report, Newzoo.*

<https://newzoo.com/solutions/standard/market-forecasts/global-games-market-report/>

[10] *Hyper-Casual Games CPI Benchmark Report - 2018, Tenjin.*

<https://cdn2.hubspot.net/hubfs/4079671/Tenjin%20Hyper-Casual%20Games%20CPI%20Benchmark%20Report%20-%202018.pdf>

[11] *Hyper-casual games: What are they & how do you monetize them?, IronSource.*

<https://www.ironsrc.com/blog/what-are-hyper-casual-games-and-how-do-you-monetize-them/>

[12] *How to make an Ultra Casual Hit game, Voodoo.*

<https://mobilefreetoplay.com/wp-content/uploads/2018/07/How-to-Make-a-Hit-Casual-Game-Voodoo-Publishing2.pdf>

[13] *5 Ways Voodoo Dominates the Hyper-Casual Market, Deconstructor Of Fun.*

<https://www.deconstructoroffun.com/blog/2018/8/1/5-ways-voodoo-dominates-the-hyper-casual-market>

[14] *The Mobile Gaming Industry: Statistics, Revenue, Demographics, More [Infographic], Mediakix.*

<https://mediakix.com/blog/mobile-gaming-industry-statistics-market-revenue/>

[15] *Deconstructing Mobile & Tablet Gaming 2017, EEDAR.*

<https://www.npd.com/lps/pdf/eedar-mobile-report-2017.pdf>

[16] *La ascendente industria chilena de videojuegos, Pauta.*

<https://www.pauta.cl/negocios/la-ascendente-industria-chilena-de-videojuegos>

[17] *Practical Game Design, Adam Kramarzewski & Ennio de Nucci*

Packt Publishing Ltd, ISBN 978-1-78712-179-9

[18] *Hunicke, Robin, Marc LeBlanc, and Robert Zubek. "MDA: A formal approach to game design and game research." Proceedings of the AAAI Workshop on Challenges in Game AI. Vol. 4. No. 1. 2004.*

[19] *López, P. (2016). Diseño e implementación de un videojuego para plataforma Android. Universidad Técnica Federico Santa María, Valparaíso.*

[20] *Manifiesto ágil.*

<https://agilemanifesto.org/>

[21] *How the kanban methodology applies to software development, Atlassian.*

<https://www.atlassian.com/agile/kanban>

[22] *Wrong Import Settings are Killing Your Unity Game, The knights of Unity.*

<http://blog.theknightsofunity.com/wrong-import-settings-killing-unity-game-part-2/>

[23] *What is the keystore, Unity Answers.*

<https://answers.unity.com/questions/1416082/what-is-the-keystore.html>

[24] *Support for Android App Bundle (AAB) in Unity 2018.3 beta, Unity Blog.*

<https://blogs.unity3d.com/2018/10/03/support-for-android-app-bundle-aab-in-unity-2018-3-beta/>

[25] *International Age Rating Coalition, IARC.*

<https://web.iarc.services.com/>

[26] *La estrategia del océano azul: cómo crear en el mercado espacios no disputados en los que la competencia sea irrelevante, Kim, W. Chan, Renée Mauborgne, and Adriana De Hassan. Grupo Editorial Norma, 2005.*

[27] *2019 Predictions: #2 the Hyper-Casual Party, as We Know It, Will End, Deconstructor of Fun.*

<https://www.deconstructoroffun.com/blog/2019/1/16/2019-predictions-2-hyper-casual-games-perfect-hybrid-monetization>

[28] *The truth about hyper-casual Gamefest 2018, IronSource.*

<https://www.ironsrc.com/wp-content/uploads/2018/10/External-The-truth-about-hyper-casual-Gamefest-2018-Omer-Kaplan-.pdf>

[29] *The rise of the jump, Polygon.*

<https://www.polygon.com/features/2014/1/20/5227582/the-rise-of-the-jump>

[30] *Privacy Policy, Paperseed.*

<https://paperseed.cl/privacy-policy/>

[31] *Loot boxes should be banned, says US senator, BBC News.*

<https://www.bbc.com/news/technology-43906306>

[32] *Un niño gasta todos los ahorros de su madre en micropagos de FIFA 18, Meristation.*

https://as.com/meristation/2017/12/20/noticias/1513749240_171754.html

[32] *From MDA to DDE, Gamasutra.*

https://www.gamasutra.com/blogs/WolfgangWalk/20151111/259078/From_MDA_to_DD E.php

[33] *Horas extra no pagadas, trabajo sin contrato y despidos masivos: el lado oscuro de la industria de videojuegos, La Tercera.*

<https://www.latercera.com/que-pasa/noticia/horas-extra-no-pagadas-trabajo/759379/>

