



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Desarrollo de un videojuego multijugador local con Unity3D: Diseño y Animación

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Jesús Lor Albert

Tutor: Patricio Letelier Torres

2016 - 2017

Resumen

El videojuego se ha convertido, durante los últimos años, en la industria del entretenimiento más potente y con más proyección de futuro. Cada vez son más los pequeños estudios que desarrollan sus ideas y se abren hueco en una industria de tal envergadura a base de pasión, creatividad y esfuerzo. Son esas características las que nos motivaron a dar forma y viabilidad económica a nuestras ideas.

Este proyecto aborda el proceso de desarrollo de un videojuego multijugador desde la perspectiva del emprendimiento, abarcando desde el nacimiento de la idea base hasta la implementación del Mínimo Producto Viable, pasando por las fases de estudio de mercado y de pruebas de los prototipos.

Palabras clave: videojuego, emprendimiento, desarrollo, estudio de mercado.

Abstract

The videogame industry has become in recent year the biggest and more promising entertainment industry. More and more small studios are starting to develop their videogame ideas, getting in on an industry of such magnitude through passion, creativity and hard work. Those are the elements that pushed us to craft and to give economic viability to our own ideas.

This paper is about the development process of one of those ideas in the form of a multiplayer videogame through the entrepreneurship perspective, from the birth of the idea to the implementation of the minimum viable product, passing through the market analysis and the prototypes testing phases.

Keywords : videogame, entrepreneurship, development, market analysis.

*“Dedicado a todos los
que me enseñaron a
no rendirme.”*

Tabla de contenidos

1. Introducción.....	8
1.1 Motivación	9
1.2 Objetivos.....	10
1.3 Estructura de la memoria.....	10
2. Concepto del juego	12
2.1 Idea inicial	12
2.2 Evolución hasta la idea final	13
3. Evaluación de la idea de negocio	16
3.1 Estado del arte.....	16
3.1.1 Un poco de historia: videojuegos orientados al multijugador local	16
3.1.2 Plataformas de venta digital	19
3.1.3 Videojuegos similares a Gnomore Gnomes	21
3.2 Lean Canvas	31
3.3 Análisis DAFO.....	32
3.4 Proyección económica	33
4. Proceso de desarrollo	35
4.1 Mapa de características	36
4.2 Cronología del proyecto.....	37
4.2.1 Experimento 1.....	40
4.2.2 Experimento 2.....	45
5. Tecnologías utilizadas.....	52
5.1 Zbrush.....	52
5.2 3D Studio Max	53
5.3 Adobe Photoshop	54
5.4 Adobe Illustrator.....	54
6. Diseño y Animación.....	56
6.1 Diseño de personajes.....	56
6.1.1 Boceto.....	56
6.1.2 Modelo 3D (Personajes: el Orco).....	57

6.1.3 Texturizado	62
6.1.4 Rigging.....	64
6.1.5 Animación.....	69
6.2 Diseño de niveles	72
6.3 Introducción al sistema de partículas.....	74
6.3.1 Ejemplo: Fuego.....	75
7. Conclusiones y trabajo futuro.....	79
Referencias y Bibliografía.....	81
Índice de Ilustraciones	83
Anexo:.....	85
Flujo de una partida en <i>Gnomore Gnomes</i>	85



1. Introducción



Figura 1: Logo del videojuego

Nos encontramos, sin duda, en la década donde los videojuegos han dejado de ser un entretenimiento digital secundario para alzarse como una de las industrias con mayor impacto económico actual. Sólo en Estados Unidos, la industria del videojuego ha pasado de generar 2 mil millones de dólares en 1998 y 6.9 mil millones en ventas en el 2005 a 17.1 mil millones en el 2010.[1]

Un factor determinante ha sido el ascenso de las plataformas de venta digital, tales como *Steam*¹, *Xbox Live*² o *Playstation Network*³. Estas plataformas han permitido que cada vez más estudios pequeños, con bajos presupuestos y personal, sean capaces de publicar y llegar al mismo número de usuarios que las grandes superproducciones, pudiendo incluso competir con ellas en cuanto a reconocimiento y ventas. Estos juegos de estudios menores son conocidos como videojuegos independientes (*indie* como término coloquial).

Este trabajo final de grado aborda el proceso de desarrollo de un videojuego independiente abarcando desde su idea inicial hasta la puesta en marcha de una campaña de financiación por *crowdfunding*⁴. Para lograr dicho objetivo, se ha dividido el proceso en cuatro etapas:

- Estudio de mercado y definición de la proyección económica del proyecto
- Desarrollo de un MVP (mínimo producto viable)
- Pruebas con *early-adopters*⁵
- Publicación de la campaña de financiación.

¹ <http://store.steampowered.com>

² <http://www.xbox.com/en-US/live>

³ <https://www.playstation.com/es-es/explore/playstation-network/>

⁴ Modelo de financiación colectiva a través de donaciones

⁵ Consumidores del producto antes de que se encuentre en el mercado

El videojuego a desarrollar tiene como título “*Gnomore Gnomes*”, y está desarrollado sobre el motor gráfico *Unity 3D*. La plataforma inicial para su publicación es *Steam*, presente en *Microsoft Windows*, *Mac OSX* y *Linux* (en el capítulo dos de esta memoria se profundiza más en la plataforma). Sin embargo, en función del buen rendimiento de la campaña de financiación, no se descarta la posterior adaptación a videoconsolas de sobremesa, como *Switch*, de *Nintendo*, o *Xbox One*, de *Microsoft*.

Este proyecto ha sido desarrollado junto a Marcos Crispín en el espacio emprendedor *Start.Inf* de la *Escuela Técnica Superior de Ingeniería Informática*. En esta memoria se tratará más específicamente el proceso de diseño y animación de los elementos que conforman el videojuego, por otra parte, en el TFG de Marcos Crispín se detallarán los aspectos asociados a la inteligencia artificial (IA).

1.1 Motivación

La principal motivación para el desarrollo de este proyecto nace del deseo de tener la posibilidad de crear nuestro propio videojuego. Deseo presente desde nuestra niñez, cuando jugábamos por primera vez a los juegos que existían por aquella época. Fue ese deseo el que nos empujó a iniciar el grado en ingeniería informática, ya que sentíamos que podíamos aunar nuestro amor por la tecnología y dar rienda suelta a nuestra creatividad para crear mundos, personajes y contagiar ese amor por los videojuegos a todo aquel que tuviera un mando en sus manos y nuestro videojuego en su pantalla.

A día de hoy aún no podemos pensar en una mejor forma de ganarse la vida.

A pesar de ser es un tipo de entretenimiento relativamente joven (los primeros videojuegos datan de principios de los años 60), los videojuegos han sufrido una evolución constante que les ha permitido diversificarse en una gran variedad de géneros. Podemos encontrar desde simuladores militares o deportivos, a aventuras de rol u orientados al juego en familia.

Por ello, las opciones eran casi ilimitadas cuando nos enfrentamos a la decisión de qué videojuego queríamos hacer. Repasando nuestras propias experiencias como usuarios habituales de videojuegos, descubrimos que a pesar del paso de los años, permanecía la afición por los comúnmente llamados “videojuegos de sofá”. Estos son aquellos juegos con un enfoque multijugador competitivo, en el que todos sus jugadores están conectados al mismo sistema de juego, ya sea un ordenador personal o una videoconsola. Por tanto, la experiencia de juego se reparte entre la que aporta el propio software, como la que se genera entre los jugadores de forma externa.

Los recuerdos de nuestras experiencias entre amigos y familiares, de las rivalidades, las burlas, la emoción de ganar y la frustración de perder una y otra vez frente al mismo amigo fue lo que nos impulsó a decantarnos por este estilo de videojuego, y así poder generar esas experiencias en todos los futuros jugadores de “*Gnomore Gnomes*”.

1.2 Objetivos

El objetivo de este TFG es desarrollar una primera versión del juego *Gnomore Gnomes*. La idea es que este MVP (Mínimo Producto Viable) nos permita publicar el videojuego en la plataforma digital *Steam* en modo *Early-Access*. Este modo de venta permite a los usuarios comprar el juego a un precio reducido, sabiendo el comprador que el software tiene partes aún por completar. Esto permite al desarrollador obtener ingresos antes de que el producto esté completamente terminado. Para lograrlo, se han marcado los siguientes objetivos:

- Desarrollar un prototipo que permita jugar una partida completa con soporte para cuatro jugadores.
- Presentar el prototipo en algún evento de prueba de videojuegos en desarrollo.
- Publicar la campaña de financiación colectiva en la plataforma *Kickstarter*, que permita cubrir los costes de producción hasta que el videojuego esté a la venta en la plataforma de venta digital *Steam*.

1.3 Estructura de la memoria

La memoria se ha estructurado de la siguiente forma:

El primer capítulo introduce el tema del trabajo fin de grado, la motivación detrás de la elección del proyecto y los objetivos a cumplir al finalizarlo.

El segundo capítulo define el concepto general del videojuego, desarrollando la evolución que sufrió desde la primera idea original hasta el concepto final sobre el que está implementado el mínimo producto viable.

El tercer capítulo comprende el análisis del videojuego desde el punto de vista de mercado. Comienza introduciendo el estado del arte, haciendo un repaso de la historia de los videojuegos de corte multijugador local con especial hincapié en algunos de los referentes actuales del género, para terminar con una tabla comparativa con el presentado en este trabajo.

A continuación se realiza la proyección económica del proyecto, estableciendo las estimaciones de los costes y de los ingresos, exponiendo el *Lean Canvas* y el análisis *DAFO* (Debilidades, Fortalezas, Amenazas y Oportunidades) referentes al proyecto.

En el cuarto capítulo se desarrolla el proceso de desarrollo del videojuego, mostrando el mapa de características del videojuego, y cómo se fueron desarrollando cronológicamente, mostrando cómo afectaron las distintas pruebas con los *early-adopters* al mapa de características y a los plazos originalmente establecidos.

El quinto capítulo analiza las tecnologías utilizadas para el desarrollo del videojuego.

En el sexto capítulo se expone cómo está enfocado el desarrollo de la inteligencia artificial de uno de los personajes del videojuego, y los sistemas que intervienen en su implementación.

El último capítulo presenta las conclusiones, así como los planes de futuro para el videojuego.

2. Concepto del juego

El videojuego que se ha desarrollado a lo largo del proyecto se titula ‘Gnomore Gnomes’, (que surge de un juego de palabras entre los términos en lengua inglesa “Gnome” y “No more”) que pertenece al género arcade⁶ de carácter multijugador local para un máximo de 4 jugadores.

Como se indica en el apartado de motivación incluido en el capítulo de introducción, la idea alrededor del videojuego nace de la voluntad de desarrollar un videojuego donde prime la diversión y la competitividad entre jugadores que se encuentren en el mismo espacio físico. Uno de los factores que se tuvo en cuenta para establecer el concepto básico del juego fue el de minimizar todo lo posible la curva de aprendizaje del usuario. Esta característica es fundamental para juegos de esta índole, pues el juego está enfocado para partidas cortas entre varios jugadores. Es más que probable que cuando el usuario propietario del videojuego realice una sesión de juego, algunos de los restantes jugadores sean nuevos y no conozcan las reglas de este, o no estén familiarizados con el sistema. Siendo la base del software las partidas competitivas cortas, si miembros del equipo no saben qué hacer, ni cómo reaccionar a lo que están viendo por pantalla, la experiencia se empobrece y resulta frustrante para todos los usuarios que se encuentren jugando (tanto nuevos como habituales).

Por tanto se buscó definir unas mecánicas de juego lo más fáciles de comprender y asimilar, así como un uso del mando de juego sencillo e intuitivo (sin por ello perder identidad o características que hagan al juego divertido y atractivo para el jugador).

2.1 Idea inicial

La idea inicial se desarrolló en base a dos pilares iniciales:

- La temática del videojuego engloba la celebración de un torneo con ambientación de fantasía medieval, donde el objetivo fuera eliminar una serie de gnomos debido a una plaga que asola el ficticio reino.
- La mecánica jugable básica se compone de una serie de cortas y variadas pruebas a lo largo del torneo, estando diseñadas todas ellas con un claro enfoque hacia la acción para varios jugadores.

Después de varias sesiones de brainstorming⁷, definimos la idea base sobre la que empezar a trabajar en profundidad las mecánicas jugables. Durante dichas sesiones, para diseñar la primera de las pruebas, se buscó reinterpretar la jugabilidad del

⁶ Género de videojuegos que sigue los principios de diseño de los antiguos videojuegos presentes en máquinas recreativas

⁷ Técnica de grupo para generar ideas en un ambiente relajado

deporte conocido como rugby, añadiendo y eliminando reglas, en favor de una jugabilidad simple y rápida.

El concepto general giraba en torno a cuatro jugadores (sin subdivisión por equipos), que debían coger un gnomo que corría suelto por el terreno de juego. Una vez uno de los jugadores tenía el control de dicho gnomo, debía mantenerlo el máximo tiempo sin que otro jugador se lo arrebatara mediante un derribo. Para obtener puntuación, el personaje portador del gnomo debía ir golpeando al mismo, para así acumular puntos. Al finalizar el tiempo reglamentario, el personaje con mayor puntuación resultaba elegido ganador.

La idea sonaba prometedora, pues cumplía los requisitos que nos impusimos al comenzar el diseño de la jugabilidad: eran unas reglas sencillas de entender, con un número de acciones posibles para el jugador mínimo, que además mantenía el estilo de jugabilidad caótica y desenfrenada que buscábamos desde el principio. Decidimos utilizarlo como punto de partida y comenzamos el desarrollo del prototipo.

2.2 Evolución hasta la idea final

Como se explicará con detenimiento a lo largo del cuarto capítulo de esta memoria, durante el comienzo del desarrollo de la primera versión del prototipo, surgieron dos importantes cuestiones acerca del videojuego necesarias de resolver antes de continuar el desarrollo:

La primera tenía relación con los recursos y el tiempo disponible, pues comenzamos a observar que probablemente no sería posible llegar al final del desarrollo del Mínimo Producto Viable (alrededor de junio del año 2017) teniendo desarrolladas un número suficiente de pruebas del torneo como para aportar suficiente variedad y rejugabilidad al título. Las opciones a elegir eran escasas. La primera y más obvia era la adquisición de más recursos. Lo mínimo para llegar al plazo con el doble de pruebas era la inclusión en el equipo de un programador y un artista más. Dado que no teníamos un horario fijo de trabajo, ya que teníamos que estar acomodando los horarios de desarrollo a los horarios de las clases de la facultad o a trabajos fuera de ella, y que no disponíamos de financiación para cubrir los costes de dos participantes más, decidimos seguir buscando otra solución.

La segunda gran cuestión estuvo muy presente a lo largo de todo el proceso de desarrollo. Durante la partida, la función del jugador que tenía el control del gnomo tenía una función clara y desafiante, pues tenía que reaccionar en base de las acciones del resto de tres jugadores, para evitar perder la posesión del gnomo que porta. Sin embargo, la función del jugador que no tiene la posesión del gnomo, era excesivamente simple, pues no tenía más que perseguir al jugador que lo portaba. Como resultado, la acción de la partida se concentraba siempre en el punto en el cual se hallara el jugador portador, estando los tres jugadores restantes siempre pegados a él. Esto provocaba una situación frustrante continua, debido a que en cuanto un nuevo jugador obtenía el gnomo, pocos segundos después lo perdía, pues los otros



tres personajes ya se encontraban muy cerca de él para derribarlo. Mientras tanto, el resto del terreno de juego permanecía vacío.

Por tanto, era importante encontrar la manera de repartir a los jugadores a lo largo del terreno de juego, y, sobre todo, de hacer sentir a los jugadores que tenían un rol diferenciado sobre la partida (como pueden ser los roles de defensa, mediocentro y delantero en un partido de fútbol tradicional), y que necesitaban plantear estrategias y tomar decisiones de equipo para obtener la victoria.

Teniendo que encontrar una solución a ambos problemas, llegamos a la conclusión de que teníamos que replantear la base jugable de las pruebas si queríamos seguir siendo dos desarrolladores y mantener el grado de calidad del producto. La solución a la que se llegó fue otorgarle más profundidad a la prueba que nos encontrábamos diseñando, aportando más reglas y acciones para los jugadores y así pasar de un sistema de varias pruebas a un partido completo como si de un deporte común se tratara.

Esta decisión suponía la pérdida de la variedad inicial, pues sólo podía jugarse un tipo de partido. Para compensar la limitación de las mecánicas básicas, se optó por ampliarla presentando una serie de escenarios donde disputar los partidos, donde no sólo la estética fuera radicalmente distinta, sino que, además, cada uno de ellos aportara mecánicas jugables únicas a la base común, haciendo que los jugadores tuvieran que variar su forma de enfrentarse al juego dependiendo de cuál fuera el escenario donde se está dando lugar la partida.

Sólo faltaba aportar profundidad a las bases jugables del 'deporte' para que se sostuviera como pilar principal del juego. Durante el desarrollo del proyecto, estas fueron variando, evolucionando o siendo descartadas (Este proceso se detalla durante el capítulo cuatro de esta memoria). A continuación se enumeran los cambios que se produjeron sobre la idea inicial, en su estado definitivo:

- Los jugadores pasan a estar repartidos en dos equipos de dos jugadores cada uno.
- La raza del personaje afecta a la jugabilidad, aportando más velocidad, más resistencia y demás habilidades especiales en función de la raza elegida.
- Se añaden dos zonas de puntuación, una a cada extremo del campo, perteneciendo una a cada equipo. La forma de puntuar consiste en llegar con el jugador que se encuentre en posesión en ese momento del gnomo a los límites de la zona del equipo contrario. De esa forma, el jugador planta el gnomo en el suelo, el equipo al que pertenece obtiene un punto, y un nuevo gnomo es liberado. El equipo con más puntuación al final de tiempo de partida especificado, resulta elegido ganador.
- Se añade un sistema de sprint, mediante el cual un jugador puede aumentar su velocidad durante un tiempo determinado, teniendo que esperar otro tiempo más prolongado para que la acción vuelva a estar disponible.

- Se añade un sistema para que dos integrantes del mismo equipo puedan intercambiar entre ellos la posesión del gnomo.
- Se añade un sistema de puntuación basado en acciones durante la partida. Para que la zona de puntuación esté habilitada para que el equipo portador del gnomo anote un punto al llegar a ella, sus integrantes han de realizar una serie de las siguientes acciones:
 - Golpear al gnomo (sólo realizable por el jugador portador del mismo)
 - Completar con éxito un pase al el compañero del equipo
 - Derribar a un jugador rival (sólo realizable por el jugador no portador del gnomo)



Figura 2: Captura de una funcionalidad con la partida final.

3. Evaluación de la idea de negocio

3.1 Estado del arte

3.1.1 Un poco de historia: videojuegos orientados al multijugador local

La modalidad multijugador en los videojuegos ha estado presente prácticamente desde el primer videojuego. Ya en 1972, el conocido juego *Pong*, desarrollado por *Atari*, permitía a dos jugadores competir entre sí. Durante los primeros años de la industria, la tecnología en los hogares no estaba aún lo suficientemente desarrollada ni era fuerte en el mercado, por lo que los videojuegos tenían su mayor público en las máquinas recreativas. Dado que las máquinas se encontraban en locales de juego, era normal que los jugadores asistieran a dichos locales en grupos, por lo que muchas de las máquinas recreativas se orientaban hacia el juego multijugador. *Double Dragon*, *Street Fighter* o *Teenage Mutant Ninja Turtles: The Arcade Game*, fueron iconos de la industria de los videojuegos durante los años 80 y 90.

Cuando aparecieron las primeras videoconsolas domésticas, se buscaba trasladar la experiencia de juego de las máquinas recreativas al salón del hogar, por lo que muchos de los juegos presentes en las consolas eran directamente versiones de los mismos juegos. Sin embargo, también se comenzó a desarrollar títulos de carácter multijugador diseñados exclusivamente para consolas domésticas y ordenadores personales conforme el mercado iba creciendo, y los videojuegos se asentaban como una forma de entretenimiento más popular.

Los videojuegos de carácter multijugador local buscaban ofrecer la mayor diversión a corto plazo y provocar reacciones de tensión y competitividad en los usuarios, que ayudaran a generar un ambiente de diversión en el espacio real donde se hallaran los jugadores. Destacaban por ser juegos de partidas de corta duración y frenéticos. Es en este tipo de videojuegos donde se explotaba el paradigma del diseño de los videojuegos que trata de ofrecer mecánicas fáciles de entender pero complejas de dominar, para que el juego fuera disfrutable para novatos como desafiante para jugadores habituales.

Los videojuegos más populares de carácter multijugador durante la década de los 90, con las videoconsolas domésticas bien asentadas en el mercado mundial (destacando la *PlayStation* de *Sony* y la *Nintendo 64* de *Nintendo*, con 102 y 38 millones de consolas vendidas, respectivamente), se clasificaban en los siguientes géneros:

Género	Máximos representantes
Simulador deportivo	<i>NBA Live, FIFA</i>
Acción	<i>GoldenEye 007, Altered Beast</i>
<i>Arcade</i> (en referencia a que era el tipo de juego presente en las máquinas recreativas)	<i>Mario Kart, Bomberman, Bubble Bobble</i>
Lucha	<i>Street Fighter, Mortal Kombat</i>

Tabla 3.1: Referentes de los distintos géneros de videojuegos clásicos

Con la llegada de la llamada sexta generación de videoconsolas, formada por la *Xbox* de *Microsoft*, la *PlayStation* de *Sony*, y la *GameCube* de *Nintendo*, la modalidad orientada al multijugador local pasó a un segundo plano. A pesar de aún contar con una importante presencia en grandes lanzamientos como *Halo: Combat Evolved*, o *Star Wars: Battlefront II*, fue durante este ciclo de videoconsolas cuando se incorporó la vertiente del multijugador a través de Internet. Esta modalidad permitía jugar con un número mayor de jugadores, que además no necesitaban estar en el mismo espacio real o conectados al mismo sistema. El videojuego a través de Internet (*online*), cambió el paradigma de diseño de los modos multijugador en los videojuegos. Si bien anteriormente se precisaba de amigos o familiares para disputar una partida a un juego multijugador, con la inclusión de *Internet*, un jugador podía jugar con otros jugadores reales estando solo. Con ello, el tiempo disponible a invertir en el juego crecía significativamente, y sus sistemas se volvían más complejos, pues la curva de aprendizaje no requería ser tan rápida.



Figura 3: partida multijugador local de *Halo: Combat Evolved*

La llegada de las videoconsolas de séptima generación (*Xbox 360* de *Microsoft* y *PlayStation 3* de *Sony*) terminaron de asentar el nuevo paradigma. El videojuego con multijugador *online* abarcaba la gran mayoría del mercado, y videojuegos que se basaban en una experiencia para un sólo jugador, como *Bioshock 2* [2], *Mass Effect 3* [3] o *Tomb Raider* [4] se veían obligados a añadir al título un modo multijugador para poder ser competitivos en un mercado donde la competición por Internet reinaba. Los

juegos más populares para estas plataformas, *Call Of Duty: Modern Warfare*, *Gears Of War*, o *Grand Theft Auto V*, resultaban casi imposibles de jugar para aquellos no familiarizados con el título. Una tarea aún más difícil para un usuario que no sea un habitual consumidor de videojuegos, como un familiar de avanzada edad.

Sin embargo, en pleno auge del multijugador en línea, la compañía nipona *Nintendo* apostó por una videoconsola cuyo pilar principal era el multijugador local. El resultado fue un éxito absoluto de ventas (convirtiéndose en la quinta videoconsola más vendida de la historia)[5]: la *Nintendo Wii*. Con el videojuego *Wii Sports* como principal abanderado del sistema, *Nintendo* buscó revitalizar el videojuego como entretenimiento social, especialmente el familiar. Gracias a los mandos simples e intuitivos y las mecánicas sencillas de sus juegos, prácticamente cualquier persona (sin importar su edad o su experiencia previa con videojuegos), era capaz de disfrutar un juego en el sistema sin apenas dificultad o esfuerzo. Simplificaciones de deportes como el tenis, los bolos o las carreras de karts, conseguían una experiencia sin igual en los comedores de las casas.



Figura 4: Videoconsola Nintendo Wii

Gracias al impacto de la *Nintendo Wii*, el asentamiento de las tiendas de software digitales y las políticas de *Microsoft* y *Sony* por fomentar el desarrollo independiente, el videojuego basado en el multijugador local ha resurgido en el resto de videoconsolas y en los ordenadores personales. Actualmente no resulta especialmente difícil encontrar videojuegos con modos para varios jugadores exclusivamente local con un número de ventas destacable y una buena valoración por parte de los analistas pertenecientes a la prensa especializada. A continuación se muestran algunos de ellos:

Título	Valoración en Metacritic ⁸
<i>Rayman Legends</i>	92 (Wii U)
<i>Nidhogg</i>	81 (PC)
<i>Towerfall Ascension</i>	87 (PlayStation 4)
<i>Overcooked</i>	81 (PC)
<i>KEEP TALKING AND NOBODY EXPLODES</i>	88 (Playstation 4)

Tabla 3.0-1: Valoración en Metacritic de videojuegos con multijugador local

⁸ Plataforma web que muestra la media de todas las valoraciones de la prensa especializada de un título concreto

3.1.2 Plataformas de venta digital

Durante la gran mayoría de la corta historia de la industria de los videojuegos (aproximadamente desde la década de los años 60 hasta hoy), la venta de videojuegos ha ido ligada al formato físico. Dependiendo de la videoconsola en cuestión, el videojuego se incluía en un cartucho (*Atari 2600*, *Nintendo GameBoy*), *cd-rom* (*Sony Playstation*), o, más recientemente, en formato DVD (*Microsoft Xbox 360*) o Blu-ray (*Sony Playstation 3*, *Microsoft Xbox One*).

Sin embargo, con el asentamiento de *Internet* en los hogares durante los inicios del siglo XXI, aparecieron dos plataformas de venta digital que cambiarían el futuro de la industria del videojuego. Estas son *Steam*, de la compañía distribuidora americana *Valve*, y *Xbox Live*, de *Microsoft*, conocida mundialmente por el desarrollo del sistema operativo para ordenadores personales *Microsoft Windows*.

Una plataforma de venta digital otorga al usuario la posibilidad de comprar software directamente a través de Internet, accediendo mediante la compra a la descarga de este. Esto no sólo provoca inmediatez, (pues el comprador no ha de desplazarse a una tienda, o esperar el envío del producto de la forma tradicional), sino que además evita los costes de fabricación del soporte físico que contendría el juego (*DVD*, *Blu-ray*, etc), embalajes, y costes de distribución. Esta consecuencia permite, sobre todo a los pequeños estudios, poder publicar y sacar a la venta videojuegos con mucho menor presupuesto, pudiendo además establecer un precio competitivo en comparación de las grandes producciones. En una tienda digital de videojuegos conviven títulos independientes con precios que suelen comprender un rango entre los 5 y los 30€, y producciones de grandes compañías desarrolladoras, cuyos títulos rondan los 60 - 70€.

Un ejemplo de éxito en los videojuegos independientes gracias a la distribución digital ha sido el título *Minecraft*, desarrollado inicialmente por el desarrollador sueco Markus Persson, que durante 2016 sobrepasó las 100 millones de copias vendidas[6].

A continuación, vamos a describir a dos de las plataformas de venta digital más importantes en el panorama actual.

3.1.2.1 Steam



Steam se presentó durante la *Game Developers Conference* ⁹ de 2002, y fue publicado en el año posterior. En un inicio, la plataforma servía como sistema de venta y de soporte para los videojuegos desarrollados por la propia compañía *Valve*. Fue en

⁹ <http://www.gdconf.com/>

2005 cuando llegó a acuerdos con desarrolladoras externas para vender sus licencias en la plataforma. Actualmente *Steam* se encuentra en los sistemas operativos para ordenador personal *Windows*, *Mac OS-X* y *Linux*, estando disponible gratuitamente en 237 países. A día de redactar esta memoria *Steam* cuenta con más de 13 millones de usuarios conectados de forma simultánea, un número aproximado de usuarios totales de 125 millones, y más de 10.000 videojuegos disponibles a la venta (sólo durante 2016 ya se pusieron a la venta 2.245 juegos diferentes). Los beneficios aproximados de *Steam* durante 2016 fueron de 3.5 billones de dólares.[6]

Las características que hacen a *Steam* ser una plataforma tan popular son la capacidad de tener toda la biblioteca de juegos en un solo lugar, la facilidad que otorga para poder jugar con amigos y los eventos de rebajas (en las rebajas pertenecientes al verano del año 2016, *Steam* generó 223.2 millones con la venta de 36.8 millones de videojuegos)[7].

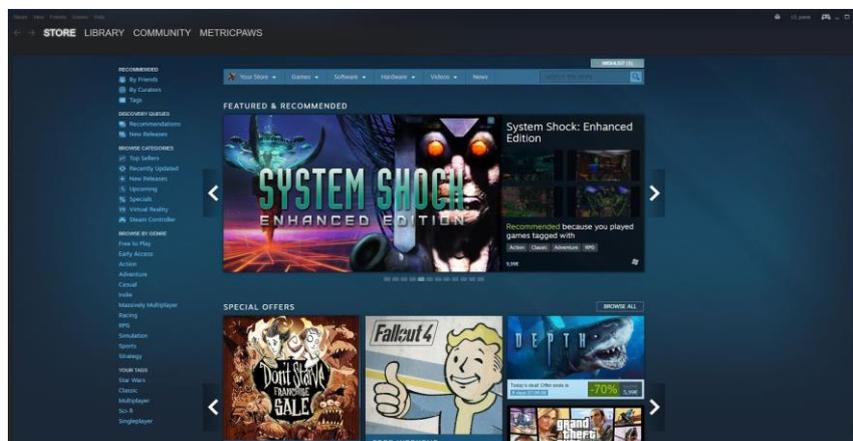


Figura 5: Captura de la interfaz principal de Steam

3.1.2.2 Xbox Live



Xbox Live es la plataforma incluida en las videoconsolas de sobremesa *Xbox* de la empresa *Microsoft*. Permite la compra de videojuegos y de contenido adicional para los mismos. Fue con el lanzamiento de su segunda videoconsola de sobremesa, la *Xbox 360*, cuando *Microsoft* apostó por el desarrollo independiente mediante la plataforma *Xbox Live Arcade*. En ella, se publicaron por primera vez grandes juegos de renombre independiente, como *Limbo* (más de 300.000 copias vendidas en su primer mes en la plataforma), o *Braid* (título que cuenta con una valoración del 95% en Metacritic, que vendió 55.000 copias durante su primera semana).

Con la reciente salida a la venta de la consola *Xbox One*, *Microsoft* reemplazó la plataforma *Xbox Live Arcade* por una versión todavía en fase temprana llamada *Xbox*

*Live Creators*¹⁰, que permitirá a cualquier desarrollador publicar su videojuego tanto en Xbox como en *Windows 10*.

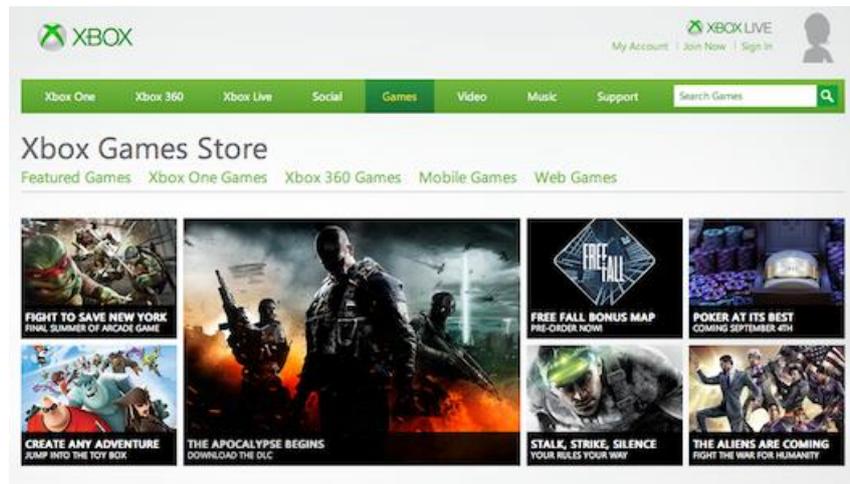


Figura 6: Captura de la interfaz principal de Xbox Live

3.1.3 Videojuegos similares a Gnomore Gnomes

Al comienzo del proceso de desarrollo del prototipo de '*Gnomore Gnomes*', realizamos una búsqueda exhaustiva de títulos orientados a varios jugadores sin conexión a Internet que gozaran de popularidad en la actualidad, y que hubieran sido publicados a lo largo de la última década (2007 - 2017). Después de probar y analizar una decena de juegos, destacamos los siguientes cuatro títulos, ya que recibieron buena aceptación de ventas por parte de los usuarios[9] y que a su vez fueron revisitados con frecuencia a lo largo del desarrollo para estudiar cómo enfocaban algunas características o funcionalidades concretas:

3.1.3.1 Towerfall¹¹

Towerfall es un videojuego independiente desarrollado en solitario por Mark Thorson que fue lanzado a la venta en 2014. Sin embargo, a día de hoy, puede encontrarse en las principales plataformas del mercado, incluyendo el ordenador personal, *Playstation 4* y *Xbox One*. Su modelo de negocio consiste en la venta del juego principal tanto en formato físico como en formato digital (con un precio de salida de 14.99\$ americanos), y de una expansión de contenidos (con un precio de 9.99\$ americanos) que añade al juego principal más escenarios, modos de juego y personajes.

Con una perspectiva en dos dimensiones y un estilo artístico "pixel art" (en homenaje a los videojuegos clásicos), *Towerfall* sumerge a los jugadores en batallas frenéticas y claustrofóbicas para un máximo de cuatro jugadores. Durante dichas batallas, que ocurren en diferentes escenarios con formas de arenas de combate (Figura 3.7), la única arma a disposición de los personajes es un arco con contadas flechas. La

¹⁰ <https://developer.microsoft.com/en-us/games/xbox/xboxlive/creator>

¹¹ <http://www.towerfall-game.com/>

variedad la aportan los escenarios, pues cada escenario introduce tipos especiales de flechas que afectan a la jugabilidad de la partida, así como una distribución diferente de los obstáculos. Los efectos especiales de las flechas propias de cada escenario incluyen, entre otros, el explotar al impactar, generar zonas tóxicas, o rebotar por las paredes, por lo que cada escenario aporta una jugabilidad radicalmente diferente a pesar de tener la misma base jugable.



Figura 7: Capturas de dos partidas en dos escenarios distintos de Towerfall

Las únicas acciones a disposición del jugador son saltar, disparar y esquivar. Y, sin embargo, debidamente utilizadas, pueden usarse de varias formas para conseguir diferentes resultados. Esa es la gran virtud del título, ya que, en palabras del periodista José Altozano, “su magia reside en que es insultantemente simple, a pesar de que maneja muchas variables y opciones. Un aspecto de su diseño muy agradecido es que absolutamente cualquiera puede entrar y tener una oportunidad.”[4]

Towerfall ofrece cuatro diferentes personajes de inicio para elegir, más cuatro personajes adicionales a desbloquear por el jugador. La elección de personaje es puramente estética, pues no tiene efecto en la jugabilidad de la partida. La expansión, llamada “*The Dark World*” incorpora un noveno personaje.

En cuanto a los escenarios, *Towerfall* fue lanzado con siete escenarios. Más tarde se actualizó de forma gratuita aportando cuatro escenarios más, esta vez teniendo que ser desbloqueados por los jugadores. La expansión, “*The Dark World*”, incorpora cuatro escenarios más. La versión para ordenador personal permite a los usuarios construir escenarios personalizados a través de una herramienta incluida en el juego.

En las partidas, diferentes modificadores aparecen de forma aleatoria, habiendo ocho tipos diferentes. Estos modificadores pueden resultar en modificadores de resistencia, velocidad, o alterar el escenario.

Towerfall se puede jugar en dos modos de juego. El primero enfrenta a un mínimo de dos jugadores, hasta un máximo de cuatro. En este modo de juego, que permite personalizar las reglas, los jugadores compiten entre ellos hasta que uno consigue un número determinado de eliminaciones, resultando ganador de la partida. El segundo modo de juego no es competitivo, si no cooperativo. De nuevo, con un mínimo y un

máximo de dos y cuatro jugadores, respectivamente, los jugadores se enfrentan a oleadas de enemigos controlados por la inteligencia artificial. El objetivo en este modo de juego es el de sobrevivir el mayor número posible de oleadas.

Towerfall tuvo una recepción positiva por parte de crítica y público, alcanzando una nota media de 87 en su versión de *PlayStation 4* en la plataforma web *Metacritic*, apareciendo así entre los 10 juegos mejor valorados en *PlayStation 4* durante su año de lanzamiento. Apareció también en la lista de los 50 mejores videojuegos de 2014[11] elegidos por los lectores de la revista especializada *Eurogamer*.

3.1.3.2 Overcooked¹²

Overcooked es un videojuego multijugador desarrollado por Ghost Town Games y distribuido por *Team17*, lanzado en 2016 para ordenador personal, *PlayStation 4* y *Xbox One*, con una versión anunciada para *Nintendo Switch*. Originalmente, el juego se vendía únicamente en formato digital. Sin embargo, con el lanzamiento de su única expansión, se lanzó a su vez una versión en formato físico para videoconsola que incluía tanto el juego principal como la expansión.

El juego, de forma alegre y desenfadada, representa el caos de gestionar la cocina de un restaurante. Los jugadores han de enfrentarse con preparar los platos que los clientes van solicitando antes de que se termine el tiempo. Para ello, deberán preparar los ingredientes, cocinarlos y servirlos en el plazo establecido. En función del rendimiento de los jugadores durante el tiempo que dura la partida, se les galardona con estrellas, siendo cero estrellas la peor puntuación, y tres la máxima.



Figura 8: Capturas de dos partidas en dos escenarios distintos de *Overcooked*

El juego principal ofrece treinta escenarios en los que jugar. Estos se disponen en serie y, para acceder a uno de ellos, se debe haber completado todos los anteriores. La expansión de contenido incluye seis escenarios adicionales.

Los personajes a elegir afectan únicamente en el campo estético, ya que todos los personajes tienen las mismas habilidades. La expansión incluye seis personajes más.

La variedad y rejugabilidad se consiguen gracias a los escenarios, ya que varían las recetas a preparar por los jugadores (como se muestra en la parte superior de la Figura 3.8), e incluyen elementos que afectan a la jugabilidad. Como ejemplo, un escenario con el suelo hecho completamente de hielo, dificulta el control de los cocineros para poder servir o manipular los ingredientes. Su jugabilidad destaca por ser extremadamente simple.

¹² <http://www.ghosttowngames.com/overcooked/>

Las únicas acciones permitidas son:

- Mover al personaje.
- Coger un objeto, como un ingrediente, plato o sartén.
- Usar un objeto, que puede ser desde cortar un ingrediente o fregar un plato, a comenzar a hervir o freír un ingrediente.

La simpleza de las acciones consiguen una curva de aprendizaje mínima. En cuestión de segundos, un jugador nuevo puede entender y jugar sin dificultad en los escenarios más básicos. La dificultad se encuentra a la hora de organizarse las tareas entre los jugadores, creando situaciones desternillantes entre los jugadores, que van pidiéndose a gritos que les traigan nuevos ingredientes o que sirvan los platos ya cocinados. Por ello es un juego que no se entiende fuera del multijugador local.

Overcooked alcanzó una nota media de 81 en la página web *Metacritic*. Fue ganador del premio a mejor debut en los *Industry Game Awards*, nominado al premio Mejor Juego Multijugador en los *Game Awards* del año 2016, así como al premio a excelencia en el diseño en el *Independent Games Festival*, entre otras nominaciones y premios.

3.1.3.3 Mario Strikers: Charged¹³

Mario Strikers: Charged es un videojuego desarrollado por el estudio *Next Level Games* y distribuido por *Nintendo*, lanzado para la videoconsola de sobremesa *Nintendo Wii* el 25 de mayo de 2007. Posteriormente fue publicado en la consola virtual *Nintendo Wii U*, videoconsola sucesora de la *Wii* original, en el año 2016. Es un juego de carácter *arcade* deportivo orientado tanto al multijugador local como al multijugador a través de Internet, para un máximo de 4 jugadores. En él, los personajes más conocidos de las diversas franquicias de la compañía de videojuegos japonesa *Nintendo*, se disputan la victoria en un partido de fútbol. Las reglas y mecánicas caóticas que se añaden a las bases del fútbol, convierte a los estadios en verdaderos campos de batalla.

A pesar de que la práctica de publicar contenido descargable adicional para los videojuegos que ya se encontraban a la venta empezaba a asentarse en la industria durante 2007, *Nintendo* mantuvo su política de videojuegos sin contenido adicional^[12] de pago hasta que la adoptó en sus videojuegos a partir del año 2014 en títulos como *Mario Kart 8*. Por tanto, *Mario Strikers: Charged*, anterior, se vendía como un producto de un único pago y todo el contenido adicional disponible era desbloqueable desde el propio juego, obteniéndose este a través de desafíos. El precio de salida fue de 39.99\$ americanos.

¹³ <https://microsite.nintendo-europe.com/mariostrikers/>

El juego se presenta con tres modos de juego:

- **Dominación:** Modo de juego en el cual el rival es controlado por inteligencia artificial, es decir, no por un jugador real. Este juego permite alterar algunas reglas de la partida, como las condiciones de victoria o la duración de la misma.
- **Copa Goleadores:** El jugador participa en 3 torneos, eligiendo previamente el capitán deseado y la dificultad. Los participantes deben superar una serie de rondas clasificatorias y eliminatorias. Al finalizar los torneos se otorgan los trofeos al equipo menos goleado y al jugador más goleador.
- **Situaciones Límite:** Es un modo de juego consistente en una serie de desafíos que, al ser completados, otorgan recompensas y elementos desbloqueables para el jugador.

Las partidas pueden disputarse en un conjunto total de diecisiete estadios:

- Cuatro estadios iniciales.
- Seis estadios desbloqueables.
- Siete estadios clásicos de ediciones del videojuego anteriores.

Los estadios desbloqueables se obtienen tras completar torneos concretos, y/o consiguiendo el máximo número de goles durante los mismos.



Figura 9: Capturas de dos partidas en dos escenarios distintos de Mario Strikers: Charged

Los personajes elegibles por los usuarios comprenden un total de doce opciones. De ellos, ocho se encuentran disponibles desde el inicio del juego, y cuatro son desbloqueados para el jugador cuando éste completa ciertos torneos. Cada uno de ellos posee una serie de habilidades especiales y tiene la capacidad de poder usar objetos para inclinar la balanza del partido a su favor, como, por ejemplo, bombas, trampas, elementos que aumentan la fuerza, la velocidad, etc. Esto amplía las

posibilidades de la partida, pues expande los conceptos básicos que presenta el fútbol, y permite al jugador experimentar con diferentes estrategias.

La idea jugable es sencilla: el equipo con el mayor número de goles en su marcador antes de que se termine el tiempo (o llegue a una determinada cantidad de goles marcados, dependiendo del modo de juego), se lleva la victoria. Otra diferencia que destaca frente al resto de videojuegos deportivos, es el balón. El comportamiento de este se ve influido por las acciones del jugador, ya que, conforme se vayan realizando acciones (tales como pases, lanzamientos o habilidades especiales) el balón brilla de forma gradual, lo que afecta a la fuerza, velocidad y precisión con la que puede ser lanzado. Al igual que en *Mario Strikers: Charged*, *Gnomore Gnomes* posee la característica de que el balón tiene un comportamiento único. En nuestro caso, el gran valor añadido es que éste se trata de un ser vivo, concretamente de un gnomo. Esto genera un cambio en las mecánicas básicas, pues los jugadores deben perseguirlo por el terreno de juego para poder hacerse con él o evitar que los jugadores contrarios lo consigan.

En conclusión, algunas de las características mejor valoradas en este título deportivo fueron la inclusión del modo multijugador online y el uso de un control más clásico que resultaba intuitivo y fácil de dominar. Debido en parte a estas características y a la cantidad y variedad de su contenido, el juego obtuvo buenas críticas, obteniendo una nota de 79 sobre 100 en *Metacritic* (basado en 47 análisis) y una muy buena aceptación por parte de los jugadores, obteniendo un 8.1 sobre 10 de nota media por parte de 127 análisis realizados por usuarios del juego. Prueba de ello es que el juego ocupa el puesto 11 de la lista de los videojuegos *Nintendo Wii* mejor valorados del año 2007. Debido a estos motivos y a la similitud en ciertos aspectos esenciales con la idea que buscábamos obtener en *Gnomore Gnomes*, fue un título que tuvimos muy en cuenta a la hora de buscar inspiración.

3.1.3.4 Stikbold!¹⁴

Stikbold! es un videojuego desarrollado por el estudio *Game Swing* que fue lanzado para las plataformas online *Playstation Network*, *Xbox Live* y *Steam* el 5 de abril de 2016. Es un juego de carácter *arcade/deportivo* orientado al juego individual o al multijugador exclusivamente local (excluyendo el modo a través de Internet) en el que pueden participar hasta un máximo de 6 jugadores simultáneos controlados bien por la computadora o por otros usuarios. En *Stikbold!*, los jugadores participan en una representación de una competición del deporte conocido en América como *dodgeball*, o más conocido en nuestro país como balón prisionero. En ella, los usuarios deben lanzarse una pelota para eliminar del terreno de juego a los contrarios y así conseguir la victoria. Los distintos escenarios aportan leves cambios en la jugabilidad básica del juego otorgando variedad al producto y aumentando su rejugabilidad.

Stikbold! posee un modelo de negocio de producto de único pago por el valor de 9,99€ a través de las plataformas online mencionadas anteriormente sin poseer ningún tipo de contenido adicional ni expansión de pago, dando la opción de adquirir, eso sí, su

¹⁴ <http://www.stikbold.com/>

banda sonora de forma totalmente gratuita. El título recibió un nuevo nivel jugable gratuito que completaba los ya existentes en su versión básica. Todo el restante contenido adicional presente en el juego puede desbloquearse jugando y nos permite obtener nuevos personajes.



Figura 10: Capturas de dos partidas en dos escenarios distintos de Stikbold!

El videojuego nos da la opción de escoger entre varios modos de juego, siendo estos los siguientes:

- **Modo Historia:** en el cual el jugador avanza a través de una aventura, compitiendo en diversos partidos de balón prisionero para avanzar por los distintos niveles. Hay que destacar que este modo de juego está enfocado únicamente a un sólo jugador y los rivales son controlados por una inteligencia artificial.
- **Equipo VS Equipo:** en este modo de juego se da a elegir entre la competición contra otros jugadores reales, o contra la inteligencia artificial. Se forman dos o tres equipos (diferenciados por el color de las equipaciones), que deben competir en un partido.
- **Todo vale:** es el clásico modo de juego conocido también como *Deathmatch* o como “todos contra todos”. En este modo, cada jugador selecciona el personaje que desee de entre todos los que ha conseguido desbloquear y pugna con el resto de jugadores por conseguir la cantidad de puntos indicados en las reglas propias de cada uno de los escenarios.
- **Minijuegos:** a parte de todos los modos nombrados anteriormente, *Stikbold!* También posee una serie de minijuegos donde el concepto jugable varía completamente, siendo una experiencia menos profunda que el modo de juego básico pero que complementa al mismo ofreciendo más posibilidades para evitar la monotonía de los modos de juego anteriores. Entre ellos podemos encontrar los siguientes:
 - **Balonmano:** la gran diferencia de este minijuego respecto al modo de juego básico es que cada equipo posee una portería propia como si de un juego de fútbol o balonmano se tratase. La forma de puntuar cambia siendo ahora el clásico sistema de goles el que ayudará a los equipos a

conseguir la victoria. Este minijuego está enfocado únicamente para 2 o 4 jugadores humanos, es decir, los equipos no pueden complementarse con personajes controlados por el ordenador.

- **Pop Pop:** en este minijuego cada jugador tiene una pelota de playa atada a uno de sus pies. El jugador alcanza la victoria evitando que desinflen su pelota y desinflando la de los demás. El escenario en el que tiene lugar este minijuego posee varios obstáculos que dificultan y otorgan cierta estrategia a la partida. De la misma manera que el minijuego anterior, Pop Pop está enfocado para 2 o 4 jugadores controlados únicamente por jugadores reales.
- **El gran escape:** en el gran escape cada jugador es poseedor de unas cajas de madera tintadas del color la equipación del personaje. La victoria en este minijuego se consigue al conseguir defender tus propias cajas y al destruir las cajas del contrario. Al igual que los otros dos minijuegos anteriores, el gran escape está enfocado para 2 o 4 jugadores controlados únicamente por jugadores reales.
- **Arcoiris a tope:** este minijuego extiende el modo de juego básico, ya que la manera de ganar es la misma, salvo que el escenario está diseñado con la intención de generar el mayor caos posible. La cantidad de obstáculos presente en él es enorme, entorpeciendo así la experiencia jugable para los usuarios. Al igual que los minijuegos anteriores, el gran escape está enfocado para 2 o 4 jugadores controlados únicamente por jugadores reales.

Los diferentes minijuegos y niveles poseen particularidades no sólo estéticas, sino que además aportan características únicas para cada uno de ellos que interfieren en la jugabilidad de la partida. De esta manera *Stikbold!* aumenta la rejugabilidad evitando que el jugador caiga en la monotonía. Además ofrece diez personajes desbloqueables para complementar a los ocho iniciales.

El juego, por lo general, obtuvo buenas críticas en la prensa especializada, logrando una valoración de 75 sobre 100 en la página web *Metacritic*. En la valoración por parte de los usuarios, este título obtuvo una nota media de 79 sobre 100. Uno de los puntos más comentados de *Stikbold!* es su intuitivo manejo, convirtiendo al título en un juego sencillo de controlar pero que exige más dedicación si se quiere dominar, sobre todo en sus niveles de dificultad más altos. Su estética humorística basada en los años 70 lo hace muy atractivo (teniendo en cuenta la simplicidad de la misma). Pero, sin duda, una de sus mayores bazas es su relación calidad/precio, ya que, sin contar con un gran número de niveles ni de modos de juego alternativos, sus 9.99€ de precio lo hacen un producto muy recomendable para pasar buenos ratos entre amigos y familiares.

3.1.3.5 Resumen

Los videojuegos anteriormente analizados nos permitieron estudiar qué características o funcionalidades eran necesarias incluir, cuáles podrían resultar interesantes, y cuáles eran prescindibles o innecesarias. Como hemos indicado a lo largo de los análisis anteriores, la inclusión de escenarios interactivos y diferenciados, era clave para conseguir un producto sólido, variado y rejugable.

A continuación, resumimos en una tabla las características presentes o ausentes en los cuatro títulos analizados, y si estas están incluidas o no en 'Gnomore Gnomes'.

					Gnomore Gnomes
▪ Tutorial	✗	✓	✓	✓	✓
▪ Modo para un jugador	✗	✗	✗	✓	✗
▪ Modo Multijugador Online	✗	✗	✓	✗	✗
▪ Modo Multijugador local	✓	✓	✓	✓	✓
▪ Número de jugadores	1-4	1-4	1-4	1-6	1-4
▪ Número de escenarios iniciales	7	30	4	6	4
▪ Número de escenarios desbloqueables	4	0	6	0	3
▪ Número de modos de juego	2	2	3	5	2
▪ Diferencias jugables entre personajes	✗	✗	✓	✗	✓
▪ Elementos estéticos desbloqueables	✗	✗	✗	✗	✓
▪ Sistema de logros	✓	✓	✗	✓	✓

					Gnomore Gnomes
▪ Sistema de Ranking	✗	✗	✗	✗	✗
▪ Sistema de perfiles de usuario	✗	✗	✗	✗	✗
▪ Soporte para Steam Workshop	✓	✗	✗	✗	✗
▪ Contenido adicional de pago	✓	✓	✗	✗	✓
▪ Plataformas	Steam, Xbox Live, Playstation Network	Steam, Xbox Live, Playstation Network	Nintendo Wii/Wii U	Steam, Xbox Live, Playstation Network	Steam, Xbox Live
▪ Precio de salida	14,99\$	16,99\$	19,99\$	9,99\$	9,99\$

Tabla 3.0-2: Tabla comparativa de los videojuegos similares a “Gnomore Gnomes”

Como podemos observar una de las grandes diferencias que aportaría *Gnomore Gnomes* sería su contenido desbloable. Tenemos pensando elaborar una gran cantidad de accesorios y aspectos distintos para cada uno de los personajes seleccionables del videojuego. Éstos tendrían un precio muy reducido y saldrían a la venta en forma de *paks* que podrían ser incluso temáticos. Otra de las virtudes de nuestro videojuego sería la jugabilidad única que aportaría cada personaje, ya que muy pocos juegos de este género en pc aportan esa característica. Por otra parte, decidimos elegir ese precio tan ajustado para el producto base, porque al ser nuestro primer videojuego en venta y no ser conocidos, pretendemos llegar a la mayor cantidad de público posible y de esta manera esperamos conseguir ese propósito. La manera de conseguir beneficios extra al precio del producto base, sería a partir de pequeñas expansiones del propio juego o de la venta de elementos personalizables como se ha mencionado anteriormente.

3.2 Lean Canvas

El Lean Canvas facilita la definición de las diversas partes de una propuesta de proyecto de cara a su presentación en una *startup*. En él se muestran, entre otros, los puntos fuertes, las debilidades, los costes, la forma de ingresos, o el mercado al cual va destinado. La siguiente tabla muestra el lean canvas que realizamos el primer día que nos embarcamos con el proyecto:

<p>2</p> <p>Problema</p> <ul style="list-style-type: none"> • La industria del videojuego se encuentra estancada en la repetición de las mismas fórmulas, tratando de replicar el éxito de juegos populares, perdiendo así el interés del jugador clásico que busca experiencias innovadoras. • La gran mayoría de los videojuegos multijugador que aparecen en el mercado están enfocados al llamado deporte electrónico, dejando de lado las experiencias que buscan la diversión sencilla entre amigos, sin exigir horas de dedicación hasta dominarlo. 	<p>4</p> <p>Solución</p> <ul style="list-style-type: none"> • Videojuego con mínima curva de aprendizaje, accesible para cualquier tipo de jugador • Mecánicas jugables únicas que no se encuentran en ningún otro título a la venta, diseñadas específicamente para el juego entre amigos / familiares. 	<p>3</p> <p>Proposición de valor</p> <p>Cuando extraes lo mejor del <i>rugby</i>, del balón prisionero, cambias el balón por un gnomo que desesperadamente trata de huir de los jugadores, todo ello concentrado en una serie de estadios locos en un mundo de fantasía, aparece el videojuego deportivo más frenético y descontrolado que más amistades dinámicas.</p>	<p>9</p> <p>Ventaja competitiva</p> <p>El coste de desarrollo del proyecto es significativamente inferior al coste estándar de desarrollo de un videojuego de sus características, permitiendo un precio de venta inferior a los juegos de la competencia.</p>	<p>1</p> <p>Clientes</p> <p>El cliente objetivo a quien va dirigido el producto es el usuario de videojuegos habitual o casual, con un rango de edad de 8 años aproximadamente e hacia delante, con afición por los videojuegos de carácter multijugador sin conexión a Internet.</p> <p>Los early adopters son los miembros de la asociación de videojuegos e-Sports Valencia y los participantes de eventos de testeo de videojuegos.</p>
--	--	---	--	---

<p>7</p> <h2>Costes</h2> <ul style="list-style-type: none"> • Horas de trabajo (aproximadamente 300h): 19.200€ • Actualización equipos: 1100€ 	<p>8</p> <h2>Métricas</h2> <ul style="list-style-type: none"> • <i>Feedback</i> en eventos de testeo de videojuegos • Nº de visitas y valoración del trailer en <i>Youtube</i> • Donación total en la campaña de financiación • Nº de descargas de la demo • Ventas en <i>Steam</i> • Jugadores/Tiempo en <i>Steam</i> 	<p>5</p> <h2>Canales</h2> <ul style="list-style-type: none"> • Los canales de comunicación serán <i>Facebook</i>, <i>Youtube</i> y <i>KickStarter</i>. • Los canales de distribución serán <i>Steam</i>, con la posibilidad de distribución en <i>Xbox Live</i> 	<p>6</p> <h2>Ingresos</h2> <p>El modelo de negocio para el producto va a ser la venta del producto mediante distribución digital en la plataforma <i>Steam</i>, con un precio de 10€ por licencia, ofreciendo además expansiones de contenido para el mismo por un coste adicional menor al del producto original (5€ por cada paquete de contenidos).</p>
--	---	--	---

3.3 Análisis DAFO

El análisis DAFO es una herramienta que resume la situación de un proyecto, mostrando las principales características propias del proyecto (fortalezas y debilidades), así como su situación frente al mercado al que se enfrenta (oportunidades y amenazas). A continuación se muestra el análisis que se realizó durante los primeros días del proceso de desarrollo:

<p>FORTALEZAS</p> <ul style="list-style-type: none"> • Precio competitivo • Mecánicas jugables únicas • Diseño original de personajes • Diseño innovador de escenarios • Bajo coste de desarrollo • Enfocado a amplio rango de edades • Sin limitación geográfica • Curva de aprendizaje mínima • Variedad en los roles jugables 	<p>DEBILIDADES</p> <ul style="list-style-type: none"> • Escaso contenido inicial • Marketing efectivo • Presencia online • Estado Financiero • Primer producto a la venta
<p>OPORTUNIDADES</p> <ul style="list-style-type: none"> • Escasas propuestas similares en el mercado • Revitalización de propuestas basadas en el multijugador local 	<p>AMENAZAS</p> <ul style="list-style-type: none"> • Saturación en el mercado de videojuegos • Competencia muy asentada

Tabla 3.0-3: Análisis DAFO del proyecto

El esfuerzo que hemos realizado para el diseño de personajes, escenarios y mecánicas es nuestra gran fortaleza. La creatividad detrás de cada elemento en el videojuego le hacen destacar frente al resto de competidores que se limitan a clonar las fórmulas y elementos exitosos. Adicionalmente, el bajo coste de desarrollo del proyecto (en comparación a los costes estándar de desarrollar un videojuego), permite un precio de venta más que competitivo.

Por otro lado, nuestras debilidades se centran en que dependemos de la campaña de *crowdfunding* para cubrir los costes, por bajos que estos sean, y que, al ser nuestro primer videojuego, carecemos de experiencia en el mercado, de reputación, y de presencia online que nos permita realizar un marketing sólido y efectivo.

En cuanto al análisis de oportunidades y amenazas, nuestra mayor oportunidad pasa por aprovechar la creciente revitalización del género del multijugador local. Sin embargo, estudios indican que la facilidad para publicar videojuegos en estos últimos años está llevando a la industria a un estado de saturación. Prueba de ello es que el 40% de los videojuegos publicados en la plataforma *Steam*, fueron lanzados durante el año 2016. Esto supone una amenaza seria para nuestro proyecto, debido a las debilidades anteriormente indicadas.

3.4 Proyección económica

Una de las partes más importantes a la hora de afrontar el desarrollo de un proyecto es la definición de la proyección económica, que estima la relación entre costes e ingresos a lo largo del tiempo de desarrollo y de la vida del producto terminado una vez se encuentre a la venta. Con estas estimaciones se puede estudiar la viabilidad de la idea, estableciendo una aproximación de en qué momento el producto comenzará a aportar beneficios, o cómo se espera que evolucione el número de clientes.

En la tabla siguiente se muestra el cálculo de la proyección económica que se realizó durante los primeros compases del proyecto. Se dividieron los intervalos de tiempo por años, estudiando la evolución durante los primeros 5, pues a partir del tercer año ya no deberían aparecer costes significativos, lo cual debería ser representativo de los beneficios que aportará el videojuego posteriormente. Los parámetros que se analizan son: número de nuevos compradores del videojuego, nuevos compradores de los contenidos adicionales del mismo, ingresos obtenidos, costes y beneficios.

Como se indica en el Lean Canvas, el precio elegido para el producto es de 10€. Este precio se estableció tras un estudio de los precios de los juegos contra los que competiría, siendo este un poco inferior a la media, dada la poca reputación del estudio y que no es un videojuego con una carga de contenido demasiado elevada en su versión básica. Esa versión podría ser ampliada con paquetes de contenido adicional, con un precio reducido (concretamente la mitad del precio de la licencia básica, 5€).



	Año 1	Año 2	Año 3	Año 4	Año 5
Nuevos compradores					
Licencia del videojuego	-	200	500	500	400
Licencia rebajada del videojuego	-	-	-	-	1000
Contenido adicional	-	-	100	150	200
Ingresos					
Licencia del videojuego	-	2000	2000	5000	4500
Contenido adicional	-	-	500	750	1000
Costos					
Personal	1.000 €	1.000 €	-	-	-
Equipos	200 €	-	1.000 €	-	-
Beneficios (Ingresos - Gastos)	-1.200 €	-200 €	1.300 €	7.050 €	12.550 €

Tabla 3.4: Proyección económica para los primeros cinco años

La estimación se ha realizado teniendo en cuenta que el producto base tendrá un valor de 9,99€ frente a los 5€ que valdría durante las rebajas de la plataforma de venta digital. Por otra parte, el contenido adicional a modo de expansiones para el videojuego también tendría un valor de 5€ para cada una de ellas.

Dado que hasta unos años el juego no generaría beneficios y que se trata de nuestro primer proyecto debemos ser cautos. No nos planteamos la posibilidad de que el videojuego sea nuestra principal fuente de ingresos, por lo que los beneficios obtenidos por el mismo serían un ingreso extra complementario a otros empleos. Para cubrir las pérdidas de los dos primeros años optamos por la opción de publicar una campaña de crowdfunding en la plataforma *kickstarter*, donde esperamos recaudar suficientes donaciones.

A partir del cuarto año, se espera un incremento de usuarios notable. Si se alcanzara un número de jugadores significativo (aproximadamente 5.000 jugadores), se estudiaría tanto lanzar el videojuego en otras plataformas, como añadir más contenido adicional. Pero, llegados a ese caso, se debería tratar como una proyección económica nueva, ya que supondría nuevos costes, y, por tanto, de necesaria financiación. En este proyecto se ha analizado la proyección económica con la idea de lanzar el juego en una sola plataforma con un sólo paquete de contenidos adicionales.

4. Proceso de desarrollo

El desarrollo de un videojuego es un proceso largo y complejo. Hasta los videojuegos en apariencia más simples conllevan un importante trabajo de diseño e implementación detrás. Cualquier mínimo fallo en la programación, o decisión de diseño imperfecta puede arruinar la experiencia jugable por completo. Para entender mejor esta cuestión, se va a tratar de ejemplificar mediante un error de programación que se produjo durante el desarrollo de *Gnomore Gnomes* que muestra cómo la atención al detalle y la organización es capital para alcanzar un producto final robusto, en los plazos previstos.

El error en cuestión se dio lugar en la acción que realizaba el lanzamiento del gnomo. El jugador que tiene el gnomo en posesión, puede lanzarlo hacia delante varios metros si así lo desea. El gnomo porta un sistema de detección de colisiones que permite que, al detectar la colisión con un obstáculo (muro del escenario u otro personaje), choque con este y así se detenga el lanzamiento. Durante el proceso de implementación se completó el código y se dejó la acción lista para realizar las pruebas. Durante ellas, se lanzó el gnomo de todas las maneras posibles, obteniendo un resultado satisfactorio. Sin embargo, unos días después, durante unas pruebas más generales de la partida, tras un lanzamiento el gnomo desapareció de los límites del terreno de juego. Esto dejaba injugable la partida, pues no había ningún sistema que controlara ese tipo de errores, y sin gnomo en el terreno de juego, la partida no puede continuar.

El problema, que no habíamos previsto, era el siguiente. La animación de lanzamiento alarga el brazo del personaje hacia delante, similar al movimiento de lanzamiento que realiza un jugador de *baseball*. Esto provocaba que, si el personaje se encontraba pegado a los límites del campo cuando el brazo estaba extendido, el gnomo (que se encuentra en ese momento en la mano del personaje) ya se encontrara fuera de los límites del campo, por lo que no se produce la detección de la colisión antes comentada, y terminaba precipitándose hacia el infinito. Esto demuestra cómo la acción más simple puede desmoronar la jugabilidad de la partida, y cómo el proceso de pruebas ha de ser más exhaustivo y profundo que en otro tipo de software.

Este tipo de situaciones puede descontrolarse si no se lleva el desarrollo del proyecto de forma metódica y planeada, definiendo qué funcionalidades se están desarrollando, cuáles están completadas, cuáles en proceso de pruebas, etc. Otro aspecto clave en la organización del desarrollo de un producto software es la gestión de las versiones. Esto comprende decidir qué funcionalidad estará incluida en cada versión, y en qué fecha estará completa cada una de ellas.

De todas las versiones por las que pasa un producto software, una de las más importantes es el llamado Mínimo Producto Viable (MVP), que corresponde a la versión que ya puede ponerse a la venta, aunque en ese momento no incluya toda la funcionalidad que portará la versión final. Por ello, en las primeras etapas del proyecto,



se definió el mapa de características, que incluye el global de funcionalidades que se incluirá en el producto, y que permite señalar cuáles de ellas estarán incluidas en el Mínimo Producto Viable, y cuáles se incorporarán más adelante. A continuación se muestra el mapa de características que definimos para el proyecto *Gnomore Gnomes*.

4.1 Mapa de características

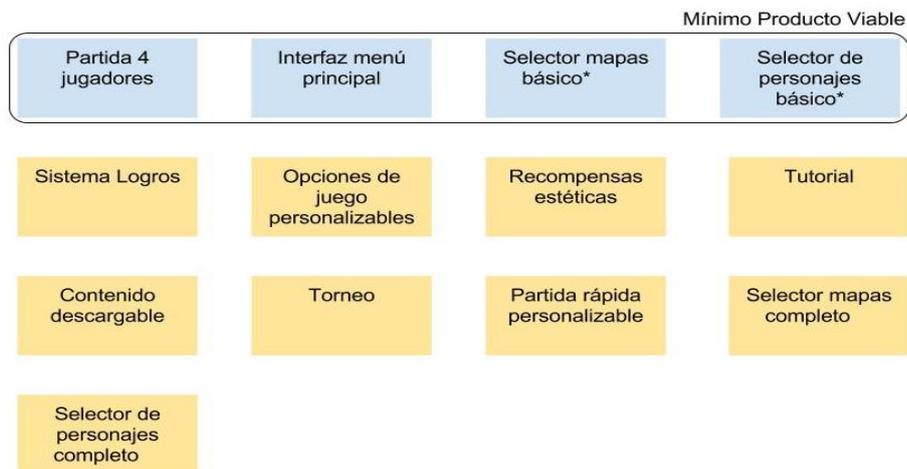


Figura 11: Mapa de características de *Gnomore Gnomes*

Una vez está definido el mapa de características, se estudia cuáles de ellas se incluirán en el Mínimo Producto Viable. Analizando los recursos y tiempo disponibles, se llegó a la conclusión de que incluiría las cuatro características marcadas en el mapa:

- Partida 4 jugadores:

Permite a cuatro jugadores competir en una partida de 5 minutos de duración repartidos en dos equipos de dos jugadores. Al final del tiempo reglamentado, se muestra qué equipo ha resultado ganador, para después volver al menú principal.

- Interfaz menú principal :

En el menú principal del MVP, se ofrece al usuario jugar una partida, entrar en las opciones, o salir del software. Un modelo de un gnomo en la parte izquierda de la pantalla indica qué opción está seleccionada. Un detalle interesante sobre el menú principal es que las opciones caen desde la parte superior de la pantalla usando el sistema de físicas en tiempo real, dando la sensación de que son bloques con peso que rebotan contra el suelo al caer.

- Selector de escenario básico

El selector de escenario básico permite al jugador seleccionar el escenario en el que se va a desarrollar la partida, por lo que es la pantalla posterior a elegir la opción 'Play'

en el menú principal. En la pantalla se muestran representaciones simplificadas de los diversos escenarios disponibles para elegir. Cuando el usuario selecciona el escenario en el que quiere jugar la partida, aparece la pantalla de selección de personajes básica.

- Selector de personajes básico

El selector de personajes incluido en el mínimo producto viable es la parte del videojuego donde cada jugador elige qué raza quiere para su personaje, así como elegir a cuál de los dos equipos disponibles quiere pertenecer (siempre que este no esté ya formado por dos jugadores). En la versión completa de esta característica, disponible en la versión final, el jugador podrá personalizar a su personaje con adornos obtenidos a lo largo de las partidas (como por ejemplo cascos, trajes o peinados), así como cambiar el color base del equipo. Una vez todos los jugadores han seleccionado la raza de su personaje, la partida para cuatro jugadores da comienzo.

4.2 Cronología del proyecto

Una vez definidas todas las características, y con el desarrollo del mínimo producto viable como objetivo, había que planear el proceso que nos llevaría hasta ello. La fecha que se estableció para tener el primer prototipo que pudiera ser probado con *early-adopters* fue enero. Encontrándonos en octubre cuando comenzamos el proyecto, era un intervalo demasiado amplio como para no subdividir el prototipo en pequeños módulos jugables que fueran integrándose conforme se desarrollaran, y así tener plazos más reducidos a lo largo de los cuatro meses. Para que la gestión de las tareas pendientes y de aquellas por completar fuera cómoda, útil y fácil de compartir, se empleó la plataforma *Trello*. En ella, definimos en forma de fichas las distintas tareas a realizar, que íbamos moviendo entre distintas secciones, y representaban funcionalidades del videojuego:

- To Do: En esta sección se encontraban las tareas que estaban pendientes de realizar. Se encontraban englobadas en dos tareas principales: “Partida mínima”, que indicaba lo que había de incluir el prototipo, y “Partida completa”, que indicaban aquellas características a implementar a partir de enero, al tener completo el prototipo.
- Design: Aquí se encontraban las tareas de tareas que se encontraban en fase de diseño. Esta es la fase en la cual se da forma a la funcionalidad perteneciente a la tarea, decidiendo en qué consiste y cómo ha de funcionar. Era muy común retornar acciones a esta fase cuando, durante las pruebas, comprobábamos que habían problemas de diseño con alguna de dichas funcionalidades.
- Doing: En esta sección aparecían las tareas de tareas que se encontraban implementándose en ese momento. Diferenciamos la sección en dos: Una para la representación artística de la funcionalidad, y otra para la programación de la



misma.

- **Test:** Cuando se hacía una versión de las implementaciones, las tareas pasaban a la sección Test, en la que hacíamos pruebas internas, comprobando que el funcionamiento fuera el deseado. Siendo un juego para cuatro jugadores y nosotros dos desarrolladores, muchas de esas pruebas eran complicadas de llevar a cabo, y no era hasta que realizábamos las pruebas con algunos amigos voluntarios, que no podíamos comprobar si la funcionalidad funcionaba correctamente en cualquier situación.
- **Done:** Aquí terminaban las tareas que recibían la aprobación tras los períodos de prueba.



Figura 12: Captura de la ficha perteneciente a la tarea de desarrollo del personaje

Cuando iniciamos el desarrollo en Octubre, definimos las dos tareas de “Partida mínima” y “Partida Completa”, y colocamos las primeras tareas en la etapa de diseño. Estas fueron las reglas básicas de una partida, el comportamiento del gnomo en el terreno de juego, y el funcionamiento de un personaje (para el primer prototipo, los personajes estarían representados todos como Orcos). Como se muestra en la siguiente captura, la tarea de “Personaje Orco” comprendía todas las acciones a realizar por un personaje.

La captura mostrada a continuación pertenece al 10 de Noviembre, cuando la tarea se encontraba en la sección “Doing (Art)”, en la cual se estaban implementando las animaciones de cada una de las acciones. Cuando las animaciones del personaje estaban completadas, se pasaba a gestionarlas a través del código en la etapa “Doing (Code)”, activando o desactivando las animaciones en función de lo que sucediera en el videojuego. El proceso de animación y de programación se muestra en profundidad en el capítulo 6 de las distintas memorias.

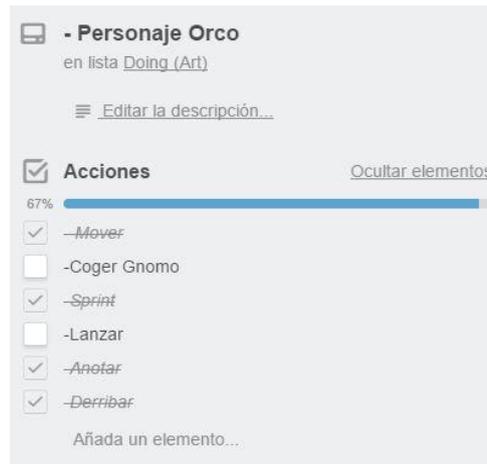


Figura 13: Captura de la ficha perteneciente a la tarea de desarrollo del personaje

El desarrollo avanzaba según lo previsto, hasta que, probando una de las acciones del personaje, esta afectaba al funcionamiento del resto, dando lugar a situaciones indeseadas en la jugabilidad. Más concretamente, era la acción de *sprint* la que provocaba colisiones con el resto de acciones. Aunque se profundiza más en esta acción más adelante en este capítulo, basta con decir que era una acción que proporcionaba un instante de aceleración en la velocidad de movimiento del personaje. Esta acción producía problemas al ser combinadas con las acciones de coger el gnomo o lanzar el mismo. Se debía a que el código perteneciente al personaje no estaba preparado para gestionar dos acciones simultáneamente, como eran, por ejemplo, realizar un *sprint* a la vez que se trata de coger el gnomo.

No hubo otra solución que replantear y reescribir todo el código del personaje, suponiendo un duro revés con los plazos previstos. El nuevo sistema de control de acciones de los personajes no estuvo listo hasta un mes después, a mediados de Diciembre, lo que obligó a retrasar la prueba con *early-adopters* de principios de enero a finales del mismo.

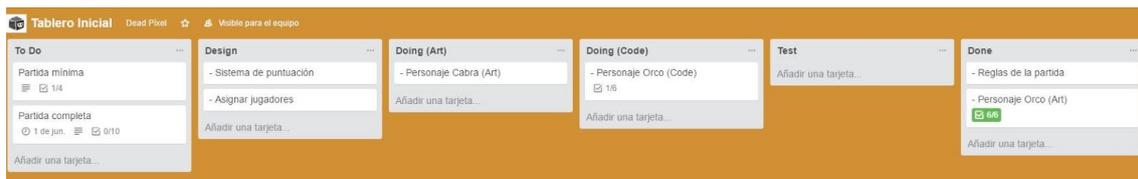


Figura 14: Captura del tablero de tareas del 19 de Diciembre

En la captura anterior se muestra el estado de las tareas, mientras se corregía el problema con el manejador de acciones del personaje. Como se puede observar, las animaciones del personaje se encontraban completas, y ya se empezaba a trabajar en las animaciones del siguiente tipo de personaje. A su vez, nuevas funcionalidades se encontraban en la etapa de diseño, y la programación del primer personaje permanecía en la sección “Doing”.

Llegado el final del mes de enero de 2017, conseguimos, según los plazos que nos habíamos establecido, tener lista una primera versión jugable con los aspectos básicos del juego en la que poder probar una partida completa entre cuatro jugadores. Como

dato curioso, el tiempo empleado en corregir el código había permitido que las animaciones del segundo tipo personaje, la cabra, estuvieran finalizadas y listas para incluir en la primera prueba (aspecto no previsto inicialmente), por lo que fue posible que durante el primer experimento, los jugadores pudieran optar entre dos tipos de personajes.

El día 20 de enero organizamos una reunión con algunos compañeros que se ofrecieron voluntariamente a formar parte de la prueba. La prueba consistiría en jugar cuatro partidas. En ellas, se aseguraría que se probarían todas las acciones posibles del juego, desde el control de menús al control de los personajes, analizando durante todo el tiempo las reacciones de los jugadores, observando el juego desde fuera (pues ninguno de nosotros dos formó parte de la prueba), recopilando información sobre fallos, detectando mecánicas que no cumplieran su propósito o directamente no resultaban estimulantes para los jugadores. Durante la primera partida, se pidió a los participantes que intentaran jugar de forma seria, tratando de lograr los objetivos que plantea el juego, es decir, tratar de ganar la partida. Una vez que comprobamos que no hubo ningún problema significativo, se dió libertad a los jugadores para que no se contuvieran y probaran cualquier acción que se les ocurriese, para así someter al juego a situaciones que no estaban específicamente tratadas o diseñadas durante el desarrollo. La prueba tuvo una duración aproximada de hora y media, contando tiempo de juego, así como tiempo entre partida y partida donde realizamos algunos cambios en parámetros del juego en función de los resultados de cada una de ellas.

En la siguiente sección del capítulo profundizamos en los resultados obtenidos durante ese primer experimento.

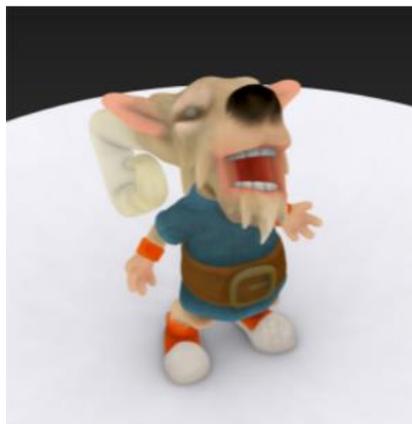


Figura 15: Captura de una de las animaciones del personaje Cabra

4.2.1 Experimento 1

Fueron las mecánicas que no resultaban estimulantes para los jugadores las que provocaron que las conclusiones que obtuvimos del primer experimento no fueran excesivamente positivas. Por un lado, el juego tenía el carisma y la originalidad que buscábamos desde el principio, lo que supuso un gran orgullo, pero ciertas mecánicas que diseñamos intentando aportar dinamismo y cooperación a la partida, resultaron

poco útiles y confusas. A continuación se analizan todas ellas, indicando por qué no daban el resultado esperado, y cómo actualizamos su diseño para darles el peso en la jugabilidad que pretendíamos inicialmente:

- Intercambio de gnomo entre dos jugadores del mismo equipo

Como ya se mencionó anteriormente, el personaje portador del gnomo es el foco principal de atención durante la partida. Los dos jugadores rivales sólo tienen como misión arrebatarse el gnomo a ese jugador, y este se encuentra en una situación de vulnerabilidad constante. Es por tanto tarea del compañero de equipo del portador el ayudar a mantener el gnomo en posesión, para así lograr llevarlo hasta la zona de puntuación.

Pues bien, para ello tuvimos que diseñar mecánicas que le permitieran ayudar a lograr ese objetivo de forma cómoda y eficaz. La primera mecánica a su disposición es común a los rivales y funcionó a la perfección durante el experimento: los jugadores que no portan el gnomo son capaces de realizar placajes que derriban al jugador con el que impactan. Sin embargo, esta acción no es siempre útil. Por ejemplo, si los dos jugadores rivales se encuentran ya cerca del portador, es difícil que el compañero sea capaz de derribar a los dos rivales antes de que estos derriben al portador.

La solución a la que llegamos fue que, si el portador del gnomo fuese capaz de pasarlo a su compañero, provocaría un cambio en la forma de jugar. Volviendo al problema antes mencionado, si dos rivales se encuentran ya cerca del portador, pero esta vez el portador pasa el gnomo al compañero desmarcado, este pasará a ser portador pero encontrándose sin oposición para avanzar hasta la zona de puntuación. Así, los dos rivales tendrían que organizarse para evitar esta situación tan desventajosa, aportando estrategias y colaboración entre jugadores (objetivo que consideramos clave para un juego de estas características).

Para esta primera versión del videojuego, el sistema de pase funcionaba de la siguiente manera: el jugador portador del gnomo, al pulsar el botón del mando correspondiente, entraba en modo lanzamiento, donde, manualmente, podía indicar la dirección en la que deseaba realizar el lanzamiento.

Una vez lanzado, si el gnomo llegaba hasta la posición del compañero, este pasaba a ser el portador de este, y el pase era completado. Si, por el contrario, el gnomo topaba con un jugador rival, el gnomo impactaba contra él y este era derribado como si de un placaje se tratara.

Pensamos que sería interesante que el sistema de pase fuera totalmente dirigido por el jugador, premiando así la habilidad de dirigir el pase en pocos segundos antes de que los rivales se encontrasen demasiado cerca. Sin embargo, el resultado obtenido no cumplía dicho objetivo. La acción en la partida era demasiado frenética, provocando que el jugador portador no dispusiera de tiempo suficiente como para poder apuntar hacia su compañero y realizar el pase.



La situación resultante solía ser que el portador realizaba el pase pero lanzando el gnomo lejos de la posición del compañero, o directamente era derribado antes de que pudiera siquiera realizar la animación de lanzar. Debido a la dificultad de la mecánica, los jugadores terminaban prescindiendo de ella, limitándose el compañero del portador a realizar placajes. El componente táctico perdía excesivo peso y el juego pasaba a ser una sucesión de derribos sin sentido.

A pesar de que la mecánica era difícil de usar de forma efectiva, decidimos conservarla para aquellos jugadores que fueran lo suficientemente habilidosos como para darle uso durante las partidas. Sin embargo, seguía siendo imperioso que hubiera una forma fácil de pasar el gnomo entre compañeros de equipo.

Barajamos varias ideas para crear una nueva mecánica de pases. La que finalmente optamos como la más correcta, fue crear un sistema de pases automáticos. El portador del gnomo, simplemente pulsando un botón, se encararía automáticamente en dirección a su compañero de equipo realizando el pase (siempre y cuando se encontrase a una determinada distancia). Del mismo modo, el personaje receptor del pase, se desplazaría de manera automática para recibir el gnomo lanzado. Durante este pequeño lapso de tiempo, el personaje que recibiera el pase, quedaría fuera del control del jugador a los mandos, aunque al ser un tiempo tan reducido (centésimas de segundo), éste era casi inapreciable por el mismo. De esta forma se garantiza que el pase siempre se produce con éxito.

Fue una decisión acertada, pues al observar los resultados en el segundo experimento que realizamos (el cual se comentará en el siguiente punto), la vertiente estratégica y de juego en equipo que buscamos se vió gratamente mejorada.

- Sprint

Otra de las mecánicas que ya se encontraban implementadas en esta primera versión jugable de *Gnomore Gnomes*, permitía a los jugadores (que no eran portadores del gnomo) la habilidad de realizar una aceleración repentina que hacía al personaje recorrer un gran tramo de terreno de juego en mucho menos tiempo que desplazándose de forma normal.

El objetivo principal, cuando decidimos introducir esta mecánica en el juego, era la de conseguir que los jugadores pudiesen evitar ser derribados, o sorprender al portador del gnomo acercándose a éste mucho más rápido. Una vez más, no obtuvimos el resultado deseado. En su lugar los jugadores abusaban del uso de la mecánica pulsando indefinidamente el botón del mando correspondiente, desde el inicio hasta el fin de la partida. Llegamos a la conclusión de que esto ocurría, en gran parte, porque era una habilidad que otorgaba ventaja en el terreno de juego, pero, sin embargo, su uso no penalizaba al jugador de ninguna manera. Si ya era especialmente difícil mantener el gnomo en posesión, desplazarse más lento que el resto de jugadores hacía la tarea de llevar el gnomo a la zona de puntuación prácticamente imposible.

La idea de que los jugadores pudieran realizar *sprints* durante el juego nos parecía atractiva, ya que aporta una capa de profundidad a la jugabilidad. Sin embargo,

después de hablar detenidamente sobre las impresiones que esta mecánica nos había producido durante las partidas de prueba, decidimos rediseñarla completamente. El efecto jugable que queríamos conseguir era que fuera una acción útil, pero que su uso descuidado supusiera una desventaja.

El nuevo diseño de la acción de *sprint* rechaza que se trate de un breve “acelerón”, coloquialmente hablando, en favor de un periodo de tiempo durante el cual la velocidad del personaje es superior. En el momento en el que un jugador pulsa el botón de *sprint* asignado, la velocidad del personaje al desplazarse es significativamente superior.

Su velocidad vuelve a su valor natural si bien el jugador deja de pulsar el botón, o un determinado tiempo es superado. Una vez llegado a ese punto, el tiempo de *sprint* permitido se va recuperando hasta llegar a un valor máximo. La velocidad a la que el tiempo permitido se recupera no será la misma a la cual se consume cuando se está realizando el *sprint*. Esa es la penalización que buscábamos conseguir dentro de la ventaja que supone estar en modo *sprint*: abusar de él puede provocar que rápidamente el tiempo permitido se consuma, y mientras el jugador espera a que el tiempo se recupere (la velocidad de recuperación es más lenta), otro jugador que ha administrado mejor su tiempo de *sprint* le supere en velocidad en alguna situación clave donde llegar primero sea importante (por ejemplo para coger el gnomo que está liberado).

Una vez diseñada la mecánica, era importante encontrar el mejor modo de indicar visualmente a los jugadores cuánto tiempo de *sprint* tenían disponible en cada momento. La solución fue la de colocar en las esquinas superiores de la pantalla dos barras en cada una (cada barra perteneciente a un jugador,) que se iban llenando o vaciando en función del uso de la acción de *sprint* que realizaban. Si un personaje tenía todo el tiempo de *sprint* para consumir, la barra aparecía completamente llena. Si, por el contrario, habían consumido todo el tiempo disponible, esta aparecía vacía. Si el tiempo se estaba recuperando, la barra se llenaba en proporción. Como añadido, se programó un sistema que permitía asignar colores al porcentaje de barra deseado. En la versión que se presentó en el segundo experimento, la barra aparecía roja de 0% al 25% completado, amarilla hasta el 75%, y verde hasta el 100%.

El hecho que los jugadores debieran preocuparse por gestionar la velocidad del personaje para obtener ventaja en el terreno de juego, produjo un incremento considerable de la carga estratégica durante las partidas y así se consiguió paliar los problemas que la primera versión del *sprint* producía en la jugabilidad.

- Zona de puntuación móvil

Inicialmente en esta primera versión de *Gnomore Gnomes*, la posición de la zona en la cual el portador del gnomo había de entrar para lograr un punto para su equipo iba variando a los largo de la partida. En el momento en el que uno de los cuatros jugadores obtuviera el control del gnomo, la zona de puntuación aparecía en una parte aleatoria del el terreno de juego. El jugador había de portar el gnomo hasta ella en un tiempo determinado, pues la zona iba empequeñeciéndose por momentos. Si el tiempo



transcurría y la zona de puntuación desaparecía debido a su pequeño tamaño, el portador perdía automáticamente la posesión del gnomo, este se liberaba y tenía que ser cogido de nuevo. Si un nuevo jugador lo cogía, la zona de puntuación volvía a cambiar de posición, y así a lo largo de la duración de la partida.

En un principio optamos por realizarlo de esta manera con la intención de que la partida se desarrollaría en todas las zonas del escenario y no sólo en ciertos puntos del mismo. El resultado conseguido con esta mecánica no era del todo malo, pero generaba confusión entre los jugadores. Al ser partidas tan rápidas y frenéticas, la posesión del gnomo pasaba de un equipo a otro constantemente, y el tiempo que tardaba el jugador portador en ver hacia qué dirección debía dirigirse con él, ya había sido derribado por un rival (debido también a los dos problemas que se acaban de analizar).

La forma en la que esta mecánica afectaba a la partida no llegó a parecernos del todo incorrecta, pero no era suficientemente sólida (al menos no el modo de juego predeterminado). Así que finalmente optamos por emplear un sistema de zonas de puntuación fija (una para cada equipo), con una función similar a una portería en fútbol, o una canasta en baloncesto. Sin embargo, buscamos la forma de aportar alguna novedad que hiciese original la manera de jugar y diferenciara a *Gnomore Gnomes* con respecto a otros videojuegos de carácter deportivo.

El sistema resultante consistió en una zona de forma rectangular marcada sobre el terreno para cada uno de los equipos. Sin embargo, queriendo dotar de profundidad a la acción de puntuar, no queríamos limitarnos a que el jugador portador llegara con el gnomo al límite de la zona de puntuación (perteneciente al equipo contrario). Queríamos que la acción de puntuar fuera la recompensa por una serie de acciones consecutivas, que culminaran con llevar el gnomo a la zona rival. Por ello, se definieron tres acciones que permitían acumular méritos para puntuar:

- Completar un pase con éxito
- Proteger al compañero portador derribando a alguno de los dos rivales
- Golpear al gnomo (acción sólo realizable por el jugador portador)

Para mostrar el progreso de realizar dichas acciones, se añadió a la interfaz de usuario durante la partida una nueva barra (esta vez una por equipo), que muestra cuántas acciones más hacen falta para optar a puntuar, pues cada vez que una acción de la anterior lista era completada, la barra se completaba un porcentaje determinado. En el momento en el que la barra esté completa, la zona de puntuación del equipo rival era habilitada para puntuar. Se añadieron detrás de la zona cuatro modelos en tres dimensiones de antorchas que ardían en llamas mientras la zona estuviera habilitada. Si el equipo logra puntuar, o si pierde la posesión del gnomo, su barra pasa a estar vacía.

La idea inicial respecto a las porterías quedó descartada para el modo de juego básico, aunque no descartamos en un futuro el incluirla de manera opcional, como un modo de juego alternativo seleccionable por los jugadores.



Figura 16: Captura de una de las partidas que tuvo lugar durante el primer experimento

Cuando concluyó el primer experimento, la planificación se adaptó para corregir todos los aspectos anteriores, por lo que algunas tareas que aparecían como ya terminadas, volvieron a la etapa de diseño, como eran las reglas básicas de la partida o la funcionalidad del personaje. Las tareas de rediseño, programación y pruebas de la funcionalidad corregida se alargaron hasta la primera semana de Marzo. Fue un proceso agotador y anímicamente más duro por tratarse de rehacer tareas en las que se había invertido ya mucho tiempo. Sin embargo, planificamos un sprint para tener las mejoras completadas el 20 de Marzo, y así fue. Una vez llegada la fecha en la que se cumplía el plazo, decidimos parar el desarrollo dos semanas, a modo de vacaciones, para recuperar fuerzas y motivación. Fue por aquellas fechas cuando oímos hablar del evento que se detalla a continuación, y que constituyó el segundo experimento.

4.2.2 Experimento 2

Una de las metas que establecimos en la planificación del proyecto fue el tener lista una versión jugable, con las mejoras introducidas después del primer experimento, para poder presentarla en alguno de los eventos de testeo de videojuegos que se producen a lo largo del año. En estos eventos se ofrece al público la posibilidad de probar videojuegos aún en fase de desarrollo, con el objetivo de obtener feedback y como posibilidad de prueba real con público real. La asociación Estudiantes de Videojuegos de Valencia (AEV) anunció un evento de estas características en la cervecería *Meltdown*, en la ciudad de Valencia. El evento se produjo el 22 de abril de 2017.

Así pues, decidimos inscribirnos para someter nuestro prototipo al juicio de público real y observar el nivel de interés suscitado por el juego. Consideramos que la forma más directa de recoger opiniones y sugerencias era a través de una corta encuesta que concentrara las opiniones que más nos interesaba recibir. Durante el evento, cuando un grupo de jugadores terminaba la partida, se les pedía si podían rellenar la corta

encuesta a través de Google Forms (llevamos al evento un dispositivo iPad con la encuesta cargada para facilitar la tarea). Todos los grupos tuvieron una respuesta positiva y realizaron de buen grado la encuesta. Cabe destacar que todos aportaron sugerencias en la última parte de la encuesta. En general, se trataban de jóvenes con una edad comprendida aproximadamente entre los 20 y los 30 años. De los once encuestados, nueve eran varones y dos, mujeres. Debido a la naturaleza del evento, los encuestados eran usuarios habituales de videojuegos. Algunos de ellos eran además desarrolladores que estaban presentando sus videojuegos durante el evento.

A continuación vamos a mostrar y comentar los resultados obtenidos y las conclusiones que obtuvimos:

1. ¿Qué escenario te ha gustado más?

La versión que presentamos en el evento Beers & Testing permitía jugar en dos escenarios. Ambos tenían características jugables únicas y diferentes entre sí. El primer escenario, llamado Mushboom Forest, incluye un sistema en el que aparecen objetos en el terreno de juego, representados como setas, que pueblan de manera aleatoria el escenario y que son activados por los jugadores al hacer contacto directo. Las setas otorgan una ventaja estratégica a los jugadores que las utilizan, como ralentizar a los enemigos cercanos, o derribarlos. Por otro lado, en el segundo escenario, llamado Rock'n Troll Hills, se generaban dinámicamente rocas gigantes que descendían por las colinas del escenario y atravesaban la zona de juego obligando a los jugadores a esquivarlas para no ser aplastados por ellas. En este último escenario, más que aportar nuevas estrategias de juego, se premiaba la habilidad de los jugadores, que debían estar atentos a más variables. Como podemos observar, la votación obtenida para cada uno de los escenarios fue muy similar, obteniendo una aceptación algo mayor para el escenario Mushboom Forest. Debido a ello, para siguientes escenarios deberíamos tener más en cuenta que resultan más atractivas las mecánicas que aportan componentes estratégicos a las partidas.

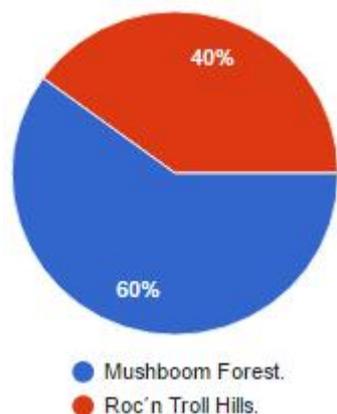


Figura 17: Resultados de la pregunta 1

2. ¿Te ha costado entender las mecánicas del juego?

Una de las mayores preocupaciones que teníamos antes de presentar el juego ante público real era que la curva de aprendizaje de nuestro juego fuese demasiado

elevada (sobre todo en este tipo de eventos, donde el tiempo que el asistente dedica a cada juego suele ser mucho menor que el de un comprador) y resultase en una experiencia frustrante. Sin embargo, gracias a los cambios realizados a partir del primer experimento, la curva de aprendizaje no fue un problema, como podemos observar en el diagrama. La gran mayoría de los jugadores indicaron entendieron perfectamente las mecánicas y las reglas del juego. Sólo un pequeño porcentaje de los jugadores indicaron que les había costado un poco terminar de comprender el objetivo. Tenemos en cuenta estos resultados y está planeado el introducir un tutorial para mostrar los conceptos básicos del juego. Adicionalmente, otra de las ideas que manejamos para mejorar la asimilación de las reglas de la partida es mostrar imágenes explicativas durante las pantallas de carga.

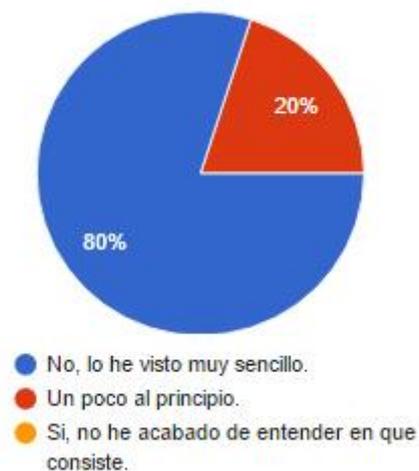


Figura 18: Resultados de la pregunta 2

3. ¿Qué valoración le darías al arte y las animaciones del juego?

El arte también fue uno de los grandes retos a los que nos enfrentamos a la hora de crear el videojuego. El diseño de personajes y escenarios es muy engañoso, pues un personaje o escenario que a vistas del desarrollador pueda parecer original y vistoso, puede no generar esa reacción en el público. Por lo tanto era de nuestro interés ver hasta qué punto nuestro esfuerzo era aceptado por jugadores. Aún siendo un prototipo, no una versión definitiva, gran parte de los elementos estéticos que se mostraban (modelos, animaciones, partículas) estaban prácticamente finalizados. La valoración del arte tuvo una aceptación bastante buena, obteniendo una única respuesta donde un usuario indicó que el apartado visual del videojuego no le había llamado especialmente la atención.

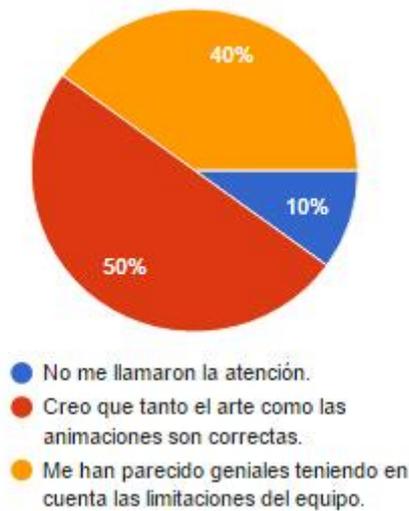


Figura 19: Resultado de la pregunta 3

4. ¿Te has sentido útil para tu equipo cuando tu compañero llevaba el Gnomo?

Uno de los problemas más latentes durante el primer experimento que realizamos en enero, fue que toda la atención de la acción que sucedía en el juego durante la partida se centraba únicamente en el jugador portador del Gnomo. Las mejoras realizadas tuvieron como cometido solucionar este problema y, como podemos observar en el gráfico, todos los encuestados afirmaron que cuando no poseían el gnomo colaboraban de distintas formas a favor de su equipo para conseguir la victoria, por lo que fueron bastante acertadas. El resultado de esta pregunta de la encuesta fue la más satisfactoria, ya que era uno de los problemas más graves que poseía la jugabilidad y en el que más tiempo invertimos para poder solucionarlo, rediseñando, implementando y probando una y otra vez las mecánicas básicas.

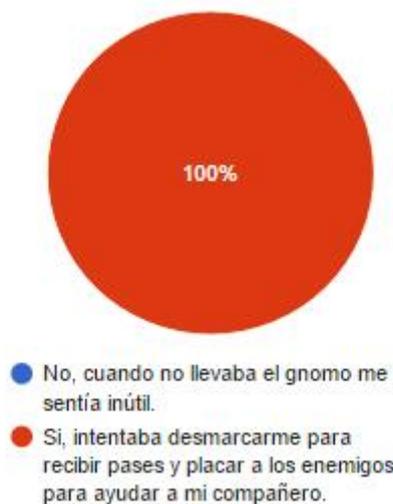


Figura 20: Resultado de la pregunta 4

5. ¿Crees 10€ por este juego sería un precio razonable cuando esté terminado e introduzca más escenarios, personajes, habilidades especiales y futuras expansiones?

Con esta pregunta pretendíamos averiguar si el modelo de negocio que habíamos elegido era factible o si por el contrario los usuarios consideraban que era un juego que debía ser gratuito. Del mismo modo también buscábamos acotar el precio de venta valorando las distintas opiniones por parte de los asistentes. Como se observa en las respuestas, un 80% del total de los encuestados opinaron que 10€ sería un precio razonable para el producto final y el resto aceptaría el hecho de que fuese algo más caro del precio que inicialmente teníamos pensado.

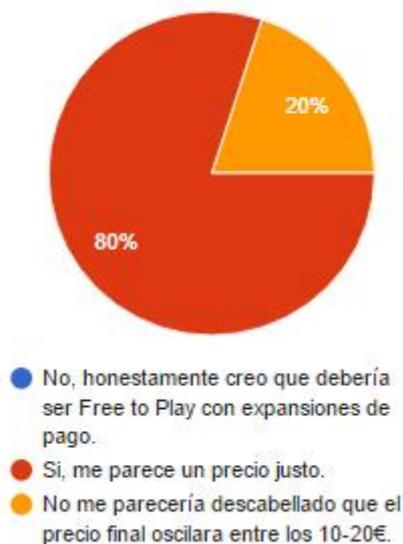


Figura 21: Resultado de la pregunta 5

6. ¿Qué valoración le darías a Gnomore Gnomes?

En la pregunta que concluía la encuesta, pedimos a los usuarios que valoraran el juego en una escala del 1 al 10 (siendo 1 la menor puntuación y 10 la mayor posible). Los resultados obtenidos fueron muy positivos, concentrándose todas las valoraciones entre el 7 y el 8 (llegando a obtener incluso una valoración de 10). Fue un buen resultado teniendo en cuenta que era un prototipo no exento de fallos y sin toda la funcionalidad implementada.

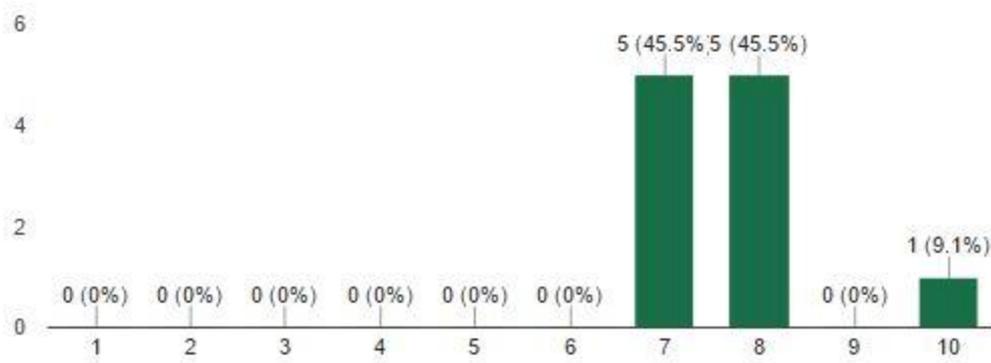


Figura 22: Resultado de la pregunta 6

7. Sugerencias u observaciones personales.

De forma opcional, dejamos un espacio al final de la encuesta donde incluir sugerencias u observaciones. Algunos de ellos se prestaron a dar su opinión y obtuvimos *feedback* valioso, con algunas propuestas muy interesantes sobre algunos elementos jugables que no nos habíamos planteado. Por ejemplo, varios usuarios insistieron en aspectos como diferenciar mejor los roles de los distintos personajes o equilibrarlos para que ninguno tuviese ventaja sobre el resto. Nos alegramos de ver que un usuario estaba interesado en el hecho de que el juego añadiera una mayor cantidad de personajes jugables y existiese la opción de poder personalizarlos, pues era una de las características que teníamos pensada incluir en el producto final. Otro usuario comentó que le había parecido confusa la elección de algunos botones del mando para ciertas acciones del personaje. Una de las sugerencias más interesantes fueron acerca de formas de mejorar algunos elementos de la interfaz de usuario.

Con el éxito durante el evento *Beers & Testing* finaliza la descripción del proceso que nos ha llevado a tener listo el mínimo producto viable. Desde este punto, hasta su puesta a la venta, están planificados tres experimentos más, siendo el tercero el más exhaustivo. En este, participaremos directamente en las pruebas, pues no serán partidas comunes las que se pondrán a prueba. En este tipo de pruebas, se realizan las diferentes acciones una y otra vez, buscando anomalías. Cada acción es posible que se haya de repetir varios cientos de veces, así como su combinación con el resto de acciones, por lo que este tipo de pruebas suele extenderse varios días.

A día de escribir esta memoria, nos encontramos trabajando en la presentación de la campaña de *crowdfunding* que nos permita financiar el resto del desarrollo del videojuego completo.

Divertido. Para familia o con amigos !
Divertido. Para familia o con amigos !
Las colisiones con otros jugadores te bloquean demasiado el movimient.
El orco podría tener alguna habilidad defensiva contra el rápido, tipo un Dash en vertical
Habilidades diferentes entre personajes, distintos tipos de pase, uso de objetos... Muy buen trabajo
Añadir mas personajes o algún estilo de personalización.
Que el mismo botón sirva para varias cosas es algo confuso en primera instancia
La barra de sprint ponerla sobre el jugador o en el círculo que lo rodea. Que se vea más claramente quien lleva el gnomo cambiando el color a parte del tamaño del círculo del jugador. Hacer el gnomo un poco más grande para que se vea mejor.

Figura 23: Resultado de la pregunta 7

5. Tecnologías utilizadas

5.1 Zbrush¹⁵



Zbrush[13] es un programa de modelado 3D, desarrollado por Pixologic, que nos permite esculpir y pintar digitalmente sobre el objeto 3D como si de un bloque de arcilla se tratase. En 1999 sus creadores dieron con este programa por *accidente* cuando trataban de buscar un software capaz de dibujar en relieve. ZBrush ha llegado a utilizarse incluso en el cine, en películas como el *Señor de los Anillos* y *Underworld*. Su versión estable más reciente es la 4R7 que data de enero del 2015 que cuenta con una versión de prueba gratuita que puede ser descargada desde la propia página de Pixologic.

El motivo principal por el que decidimos utilizar Zbrush fue porque funciona bajo el concepto de *arcilla digital*, por lo que es muy sencillo e intuitivo de aprender a manejar. Posee herramientas de bocetado tales como el uso de *ZSpheres*, donde podemos tener una aproximación de la estructura del modelo 3D que buscamos. Este programa nos permite detallar a un nivel muy superior al resto de programas similares. Los modelos en tres dimensiones que generamos, en ocasiones pueden ser una pega cuando están destinados a un videojuego, ya que una de las cosas más prioritarias en este tipo de software es la optimización y la baja carga de polígonos. No obstante, Zbrush posee herramientas de retopología como *Zremesher*, que nos ofrece un control total sobre el número de polígonos de nuestro modelo antes de exportarlo, pudiendo aumentarlo o reducirlo a elección del usuario.

El proceso de texturizado 3D también fue uno de los motivos que nos hizo elegir este programa frente a otros similares. Zbrush nos permite pintar sobre la propia geometría 3D sin tener que preocuparnos de las coordenadas de mapeado (UVW) del modelo y exportar la textura en un formato de imagen utilizable en programas externos para así poder realizar correcciones o mejoras sobre la propia textura.

¹⁵ <http://pixologic.com/>

5.2 3D Studio Max¹⁶



3D Studio Max[14] es un programa desarrollado por *Autodesk* que proporciona una solución completa al modelado, simulación, renderizado y animación 3D. Su primera versión salió para el sistema operativo DOS, en 1990. *3D Studio Max* ha sido utilizado tanto en cine, en anuncios de televisión y en muchos proyectos dentro del mundo de los videojuegos. Algunos de esos videojuegos son títulos tan importantes en la industria reciente como *Tomb Raider*, *Splinter Cell*, *Gears of War* o *Skyrim*. Su versión más reciente es la 2018 y existe una versión de prueba gratuita desde la web de Autodesk. También cuenta con una versión gratuita para estudiantes.

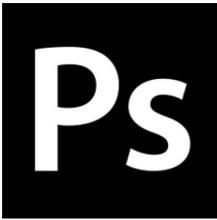
Entre las múltiples opciones que nos brinda *3DS Max*, nos ofrece la posibilidad de crear modelos 3D a partir de geometrías básicas, como esferas, cubos, etc, u otras más complejas obtenidas a partir del uso de plugins a modo de modificadores que podemos aplicar sobre las mismas. Esta función ha resultado muy útil en nuestro proyecto a la hora de crear objetos que no requerían tanto detalle, como los elementos de la escenografía o el público de los estadios.

El programa nos también permite realizar rigging sobre nuestros modelos. El *rigging*[10] es el proceso de dotar de un esqueleto con sus cadenas de huesos a los objetos o personajes 3D colocando una geometría especial a modo de huesos por debajo de la malla de los mismos para que, al ser animados, la propia malla se deforme en base a ese esqueleto.

3D Studio nos proporciona un control total sobre la animación de cada uno de los huesos que forman el esqueleto, pudiendo así ver las curvas generadas por dichas animaciones de forma gráfica y corregirlas incluso mediante parámetros con el objetivo de crear animaciones cíclicas (que se repiten constantemente), tales como caminar, correr, etc. Este tipo de animación es muy utilizada en el mundo de los videojuegos. Estas herramientas y posibilidades que nos ofrece *3D Studio* fueron determinantes a la hora de decidir usarlo en nuestro proyecto.

¹⁶ <https://www.autodesk.es>

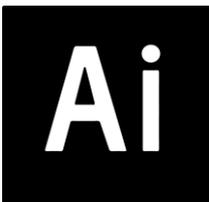
5.3 Adobe Photoshop¹⁷



Una de las características principales o que más distinguen a Photoshop es la posibilidad de usar *capas*, cada una de las cuales puede ser tratada de manera independiente dando lugar a una única imagen en el resultado final. Esta característica fue una de las opciones de peso que tuvimos en cuenta a la hora de optar por usar este programa, ya que a algunas texturas, como por ejemplo la de los personajes, se le añadieron efectos de suciedad, sombreados, correcciones de color, etc, para que tuvieran un mejor aspecto que el que obtuvimos cuando pintamos desde *Zbrush*. También nos permitió generar imágenes con canal alpha, y así dotar de zonas transparentes a las mismas, que utilizamos para generar las texturas de la vegetación y algunas banderas con rasgaduras y agujeros.

Photoshop también posee una galería con multitud de efectos que no sólo nos permite retocar imágenes ya creadas, si no también crear algunas desde cero, como fue en el caso de la textura del césped del campo.

5.4 Adobe Illustrator¹⁸



Adobe Illustrator es un editor de gráficos vectoriales en el cual se trabaja sobre un tablero de dibujo que el mismo programa denomina *mesa de trabajo*. Está destinado principalmente a la imagen y arte digital haciendo uso de dibujo vectorial. Su uso está muy difundido en varios ámbitos, tales como la generación de gráficos para plataformas móviles, impresión, publicaciones web, etc. Illustrator inicialmente sólo era compatible con Macintosh, pero a día de hoy cuenta también con compatibilidad para Windows, su última versión estable, al igual que la de Adobe Photoshop es la CC2017.

La generación de imágenes vectoriales desde este programa nos fue de gran ayuda a la hora de crear algunos elementos como los de la propia interfaz de *Gnomore Gnomes*. Las imágenes vectoriales poseen la ventaja de que no pierden calidad al

¹⁷ <http://www.adobe.com>

¹⁸ <http://www.adobe.com>

aumentar o reducir su tamaño, por lo que inicialmente al no tener muy claro ni la posición ni el tamaño de dichos elementos mencionados de la interfaz nos resultó muy útil, además su interfaz es bastante intuitiva y posee herramientas no demasiado complicadas de manejar.

Otra de las características de las imágenes vectoriales, obviamente, es que estas están formadas por vectores, de esta forma podemos corregir las mismas de manera independiente modificando su estructura a partir de unos ayudantes que nos permiten variar la misma a partir de curvas bezier, que dan flexibilidad a la imagen y obteniendo así, de forma más fiel y precisa, la imagen a la que queríamos aproximarnos.

Toda la suite de Adobe[15], compuesta por los diferentes programas de la compañía, posee versiones de prueba gratuitas que pueden descargarse de su propia web.

6. Diseño y Animación

6.1 Diseño de personajes

6.1.1 Boceto

Inicialmente, *Gnomore Gnomes* fue, como muchos proyectos, una idea muy distinta a la que acabamos desarrollando finalmente. Después de varias reuniones de *Brainstorming* y de realizar varias pruebas sobre el papel llegamos a la conclusión de que no queríamos encasillar el juego de manera que únicamente fuese atractivo a cierto tipo de público. Así pues decidimos crear en cierto modo nuestros propios personajes inspirándonos en animales, teniendo en siempre en cuenta que nos encontrábamos ante un videojuego de carácter deportivo, lo cual queríamos dejar reflejado en varios elementos, entre ellos las vestimentas de los protagonistas.

Las diferencias físicas entre ellos debían ser significativas, ya que no iban a ser únicamente estéticas, si no que cada uno de los personajes iba a tener peculiaridades físicas que influirían en sus atributos dentro del juego, así como la velocidad, la fuerza, la distancia de pase, la resistencia, etc.



Figura 24: Bocetos iniciales de los personajes

A continuación pasaremos a mostrar el proceso creativo y de diseño que seguimos para desarrollar los personajes que aparecen en el juego *Gnomore Gnomes* desde el principio hasta el resultado final obtenido después de todo el proceso, el cual será explicado detalladamente a continuación en el documento.

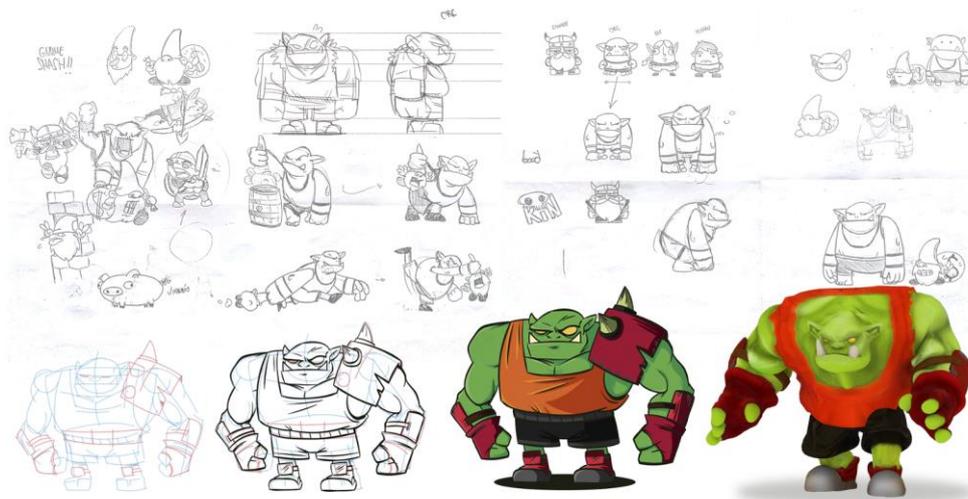


Figura 25: Evolución conceptual de uno de los personajes.

6.1.2 Modelo 3D (Personajes: el Orco)

Una vez consensuado el aspecto aproximado de cada uno de los personajes gracias a los bocetos y con la aprobación de ambos componentes del grupo en las sesiones de *Brainstorming*, llegó el momento de darle *volumen* a la idea. Para ello, inicialmente, decidimos tomar una primera opción que resultó ser no demasiado acertada. Optamos por trabajar directamente desde *3D Studio Max*, partiendo de geometría básica e ir otorgándole más detalle al personaje a medida que avanzaba su diseño.

A partir de un simple cubo, *3D Studio Max* nos permite ir subdividiendo la geometría del mismo e ir corrigiendo la posición de cada uno de los vértices, aristas y caras que lo componen hasta obtener el resultado deseado (estas posibilidades nos las proporciona uno de los modificadores, en concreto *edit poly*). Además, *3D Studio* posee un gran listado de *plugins* que el propio programa denomina *modificadores*, los cuales nos permiten interferir en la forma de los modelos permitiendo desde retorcer la geometría hasta derretirla como si de un bloque de hielo se tratase.

El resultado que empezábamos a obtener no era malo del todo, pero no se acercaba a la calidad ofrecida hoy en día por la mayoría de juegos del mercado, ya que la apariencia resultante del personaje estaba quedando demasiado poligonal.

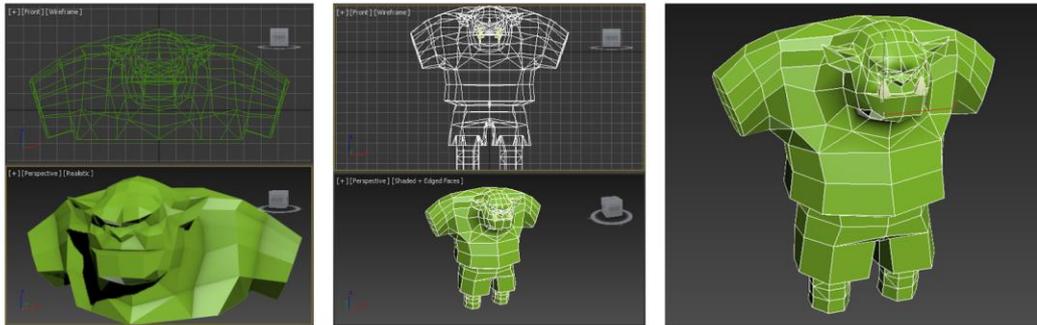


Figura 26: Proceso inicial de modelado 3D del personaje.

Así pues, decidimos cambiar de rumbo y optamos por *Zbrush*, de esta manera podríamos otorgar más detalle a los modelos de nuestros personajes. *Zbrush* posee una herramienta llamada *Zspheres*. Esta herramienta nos permite crear una estructura básica a partir de una geometría básica formada a partir de unas esferas especiales que se conectan entre sí y nos permiten moverlas, rotarlas o escalarlas en un entorno tridimensional hasta conseguir la forma aproximada del modelo que buscamos. Otra de las ventajas que nos ofrece esta herramienta de *Zbrush* es la posibilidad de usar simetría, de esta forma el modelo del personaje queda totalmente homogéneo y equilibrado. Al usar la opción de simetría, podemos crear las extremidades opuestas del personaje de manera simultánea, asegurándonos de que ambas van a resultar completamente idénticas entre ellas.



Figura 27: Vista de la herramienta Zspheres en la interfaz de Zbrush.

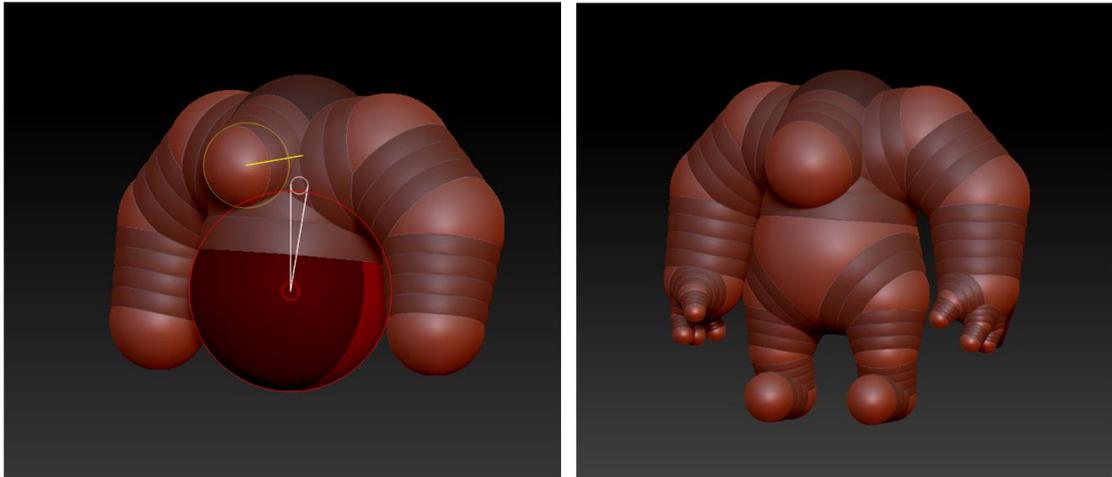


Figura 28: Generación de la estructura del personaje con la herramienta Zspheres.

Una vez tenemos la estructura básica del personaje, generada desde *Zspheres*, con cada una de sus distintas partes bien situadas, es el momento de generar la malla sobre la que vamos a trabajar. Para ellos *Zbrush* dispone de una herramienta llamada *Adaptive Skin* cuya función es recubrir por completo la estructura generada por las *ZSpheres* por una malla a modo de piel.

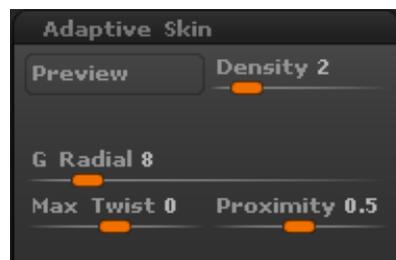


Figura 29: Vista de la herramienta Adaptive Skin en la interfaz de Zbrush.

De esta forma obtenemos una estructura similar a una escultura de *arcilla*, salvo que es digital. Sobre ella, podemos ir obteniendo la forma final deseada con la ayuda de las distintas herramientas de moldeado que nos ofrece *Zbrush*. Es un proceso laborioso que precisó de la consulta de varios tutoriales alojados en la propia web de Pixologic, puesto que el programa posee una cantidad muy amplia de herramientas para moldear que nos permiten desde añadir más arcilla sobre las partes del modelo que deseemos, inflar o desinflar ciertas zonas, hasta afilar las distintas partes de la malla para destacar zonas concretas de la misma.



Figura 30: Modelo 3D resultante al aplicar Adaptive Skin.

Tras varios días de aprendizaje con el programa y varias pruebas fallidas a modo de pruebas, llegamos a dominar la herramienta al nivel que necesitábamos para poder obtener el resultado deseado de nuestro primer personaje. De momento era sólo la malla del mismo, no poseía ni textura ni esqueleto, no obstante estaba formado por una cantidad excesiva de polígonos, pues *Zbrush* trabaja con millones de ellos para representar por pantalla el modelo con el que estamos trabajando. Esto es recomendable cuando el modelo que hemos creado está destinado para el cine, ya que al poseer tantos polígonos la cantidad de detalle del modelo 3D es muy superior. No obstante, cuando estamos hablando de un videojuego, la optimización es un factor clave. Esto es debido a que los cálculos se producen en tiempo real, y cualquier ralentización produce una experiencia defectuosa. Por tanto, reducir el número de cálculos por fotograma es capital. Desde el punto de vista de los modelos, esto se consigue reduciendo la carga poligonal lo máximo posible, sin perder un nivel de detalle significativo.



Figura 31: Modelo 3D en la fase final de modelado desde Zbrush.

A día de hoy, en un videojuego moderno los protagonistas pueden superar incluso los 100.000 polígonos, como en el caso del personaje de *Lara Croft* de la saga *Tomb*

Raider, que pasó de tener 540 en su primera aparición en 1996 en el videojuego *Tomb Raider* a más de 200.000 polígonos en uno de sus últimos juegos (*Rise of Tomb Raider* 2015).



Figura 32: Evolución gráfica del personaje *Lara Croft* de la saga *Tomb Raider*.

Uno de los objetivos que nos planteamos para los personajes de nuestro juego, fue que tuviesen una calidad gráfica aceptable en comparación con los títulos actuales en el mercado, pero la cámara salvo algunas ocasiones, en las que se acercaba durante el juego, debía estar alejada para mostrar toda la acción que ocurría en el campo junto a los personajes de los demás jugadores, el escenario, el público, etc. Así que optamos por reducir la cuenta poligonal de los personajes principales a unos 30.000 aproximadamente y de esta manera asegurarnos de que el juego funcionase fluido en un número mayor de ordenadores. Para ello *Zbrush* posee un *plugin* llamado *decimation Master*.



Figura 33: Vista de la herramienta *Decimation Master* del programa *Zbrush*.

Este *plugin* viene integrado en la versión 4R7 de *Zbrush* y nos ofrece un control total sobre la cantidad de polígonos de nuestro personaje. *Decimation Master* hace un proceso de reconstrucción (también conocido como retopología) sobre la malla del

modelo y nos permite indicar el porcentaje de polígonos que queremos llegar a reducir respecto al 100% que posee el modelo original. También de forma más precisa, podemos indicar directamente la cantidad exacta de polígonos que deseamos. Para ello primero tuvimos que pre-calcular el proceso de retopología desde la opción *Pre-process* y posteriormente le indicamos el valor aproximado de polígonos que queríamos que tuviese el personaje resultante tras la optimización. Decir que el cálculo previo a la acción de retopología es un proceso lento que dura unos minutos, pero una vez ha terminado podemos variar dicho valor y ver casi en tiempo real el resultado deseado.

A continuación mostramos una imagen comparativa del mismo modelo con distinta cuenta poligonal, para observar hasta qué punto tenemos el control del modelo resultante con esta herramienta:



Figura 34: Imagen comparative del modelo 3D con distinta carga poligonal.

A la izquierda el modelo del personaje con unos 820 polígonos frente a los 35.800 que posee el modelo de la derecha.

6.1.3 Texturizado

Una textura es una imagen de mapa de bits que se usa para cubrir la superficie del objeto virtual, ya sea 3D o 2D. El programa donde visualizamos el modelo otorga al mismo de unas coordenadas de mapeado llamadas coordenadas UVW.

Para obtener la textura con las coordenadas, tenemos que someter al modelo a un proceso de *Unwrapping*, este proceso extiende la malla del modelo sobre un plano 2d teniendo en cuenta las coordenadas del mismo y genera un mapa de bits sobre el que podemos dibujar la textura deseada. *Zbrush* cuenta con un *plugin* llamado *UV Master* que permite obtener el mapa de coordenadas y da un paso más allá, permitiéndonos pintar sobre el propio modelo como si de una figura se tratase. Para ello debemos cambiar el modo *edición* del programa a modo *dibujo*, de esta manera *Zbrush* nos da acceso a una gran cantidad de herramientas como pinceles configurables, una paleta de colores de 32 bits, etc.

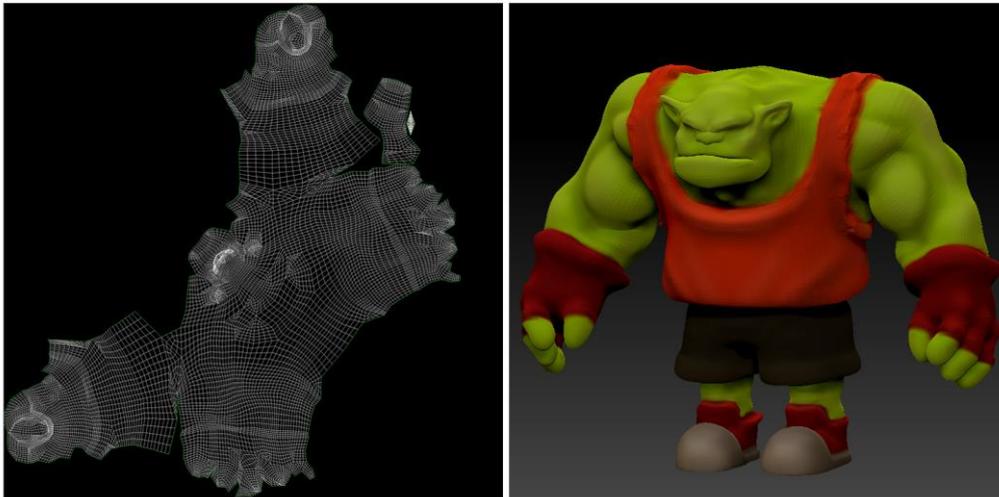


Figura 35: Obtención del mapa de coordenadas desde *Zbrush*.

Una vez tenemos el modelo 3D totalmente pintado, el programa nos deja exportar tanto el propio modelo (con las coordenadas UVW guardadas) como la textura que hemos pintado sobre el personaje, en un formato de imagen reconocible para cualquier otro programa de edición, como PNG, JPG, BMP o TIFF.



Figura 36: Vista de la herramienta exportar textura de *Zbrush*.

Aunque *Zbrush* nos ofrece múltiples herramientas para pintar nuestros modelos, decidimos cargar la textura obtenida en *Adobe Photoshop*, ya que para este ámbito es más potente, y así dotar de mayor detalle la textura y realizar correcciones sobre las costuras del mapa de coordenadas, ajustes de color, etc. De esta manera aplicamos detalles como pliegues en la ropa, arrugas en la piel, venas, remarcar los músculos, suciedad, entre otros. Cada uno de los detalles nombrados anteriormente fueron añadidos de manera independiente a la textura, ya que *Photoshop* nos permite el uso de distintas capas para no tener que *ensuciar* la imagen original de la textura. También podemos variar la opacidad de cada una de ellas o incluso el tipo de fusión de las mismas sobre el resto, lo cual nos resultó muy útil a la hora de aplicar sombreados y brillos.

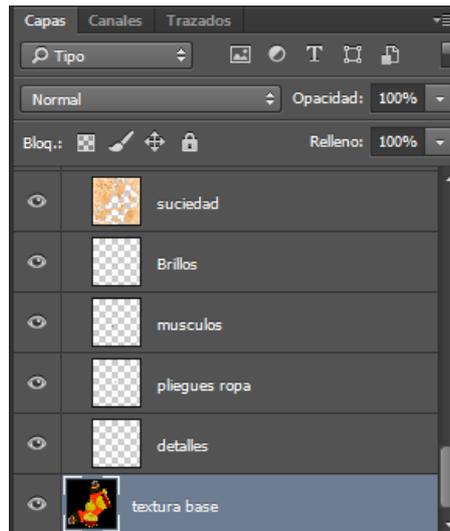


Figura 37: Esquema de las capas que forman la textura en *Photoshop*.

A continuación se muestra el resultado obtenido desde que exportamos la primera textura desde *Zbrush* hasta el resultado final después de tratar la imagen en *Adobe Photoshop*:

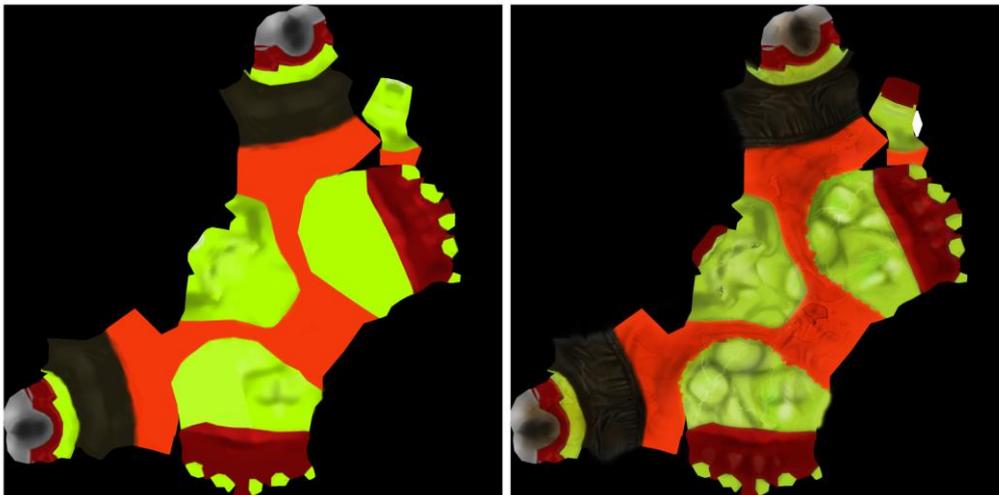


Figura 38: Resultado de la textura tras el retoque aplicado en *Adobe Photoshop*.

6.1.4 Rigging

Una vez ya tuvimos el modelo listo y con su textura correspondiente, dimos el siguiente paso y nos decidimos a animarlo. Para ello tuvimos que asignarle un esqueleto. Este proceso es conocido como *rigging*. El *rigging* es el proceso de crear un sistema de controles digitales y agregarlos a un modelo 3D para que así pueda ser animado fácilmente y eficientemente [16]. Este es un paso crucial dentro del proceso de la creación de una animación 3D.

Dentro de *3D Studio* tenemos varias maneras de crear un esqueleto para nuestros

modelos, dentro de las que conocemos, una de ellas es ir creando con geometría básica cada uno de los huesos que queremos que compongan dicho esqueleto y crear las dependencias de herencia entre ellos; Otra es crear directamente objetos del tipo hueso, en *3D Studio*, desde la pestaña de sistemas e ir definiendo *IK solvers* entre ellos, los *IK solvers* actúan a modo de tendones sobre los huesos. Esta técnica se denomina *cinemática inversa* y nos permite determinar el comportamiento o el movimiento que tendrán las articulaciones que se encuentran conectadas por sí mismas. De esta manera el movimiento queda limitado dentro de las conexiones que realicemos entre las mismas y según sus dependencias, los huesos se ven afectados unos con otros como puede verse en la siguiente imagen:

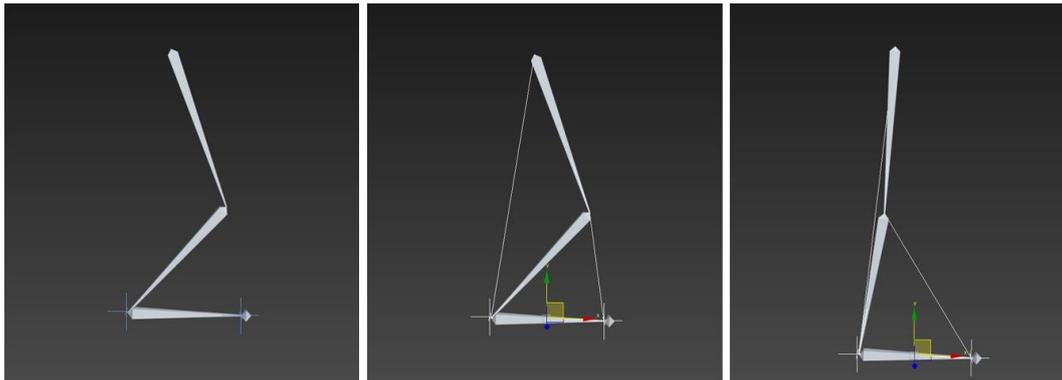


Figura 39: Sistema de huesos generados desde *3D Studio Max*.

Por último tenemos otra herramienta en *3D Studio Max* llamada *Biped*. *Biped* es un esqueleto ya creado con las dependencias entre sus huesos ya definidas. Una de las pegas que posee *biped*, como su propio nombre indica, es que está enfocado únicamente como un esqueleto para personajes que sean bípedos o humanoides, lo cual no supuso un problema para nosotros, ya que todos nuestros personajes lo eran. La herramienta nos ofrece opciones de personalización que nos permiten modificar su estructura, por ejemplo, podemos variar el número de dedos y las falanges que componen los mismos, el número de huesos que forman la columna, incluso nos proporciona la opción de generar huesos para crear una cola.



Figura 40: Vista del esqueleto de *Biped* en *3D Studio Max*.

Al final optamos por la última opción para colocar el esqueleto a nuestro modelo, pero nos surgió un nuevo problema. También queríamos que nuestros protagonistas tuvieran ciertas animaciones faciales, como abrir y cerrar la boca o levantar las cejas, así que tuvimos que crear unos huesos por nuestra cuenta e integrarlos dentro de las dependencias del propio esqueleto *Biped*.

Para ello, generamos un par de huesos para la mandíbula y uno para cada ceja y accedimos a la tabla de dependencias de *Biped* para integrarlas dentro del hueso de la cabeza. De esta manera pretendíamos conseguir que todos estos huesos se viesen afectados y se moviesen conjuntamente por el hueso de la cabeza del esqueleto.

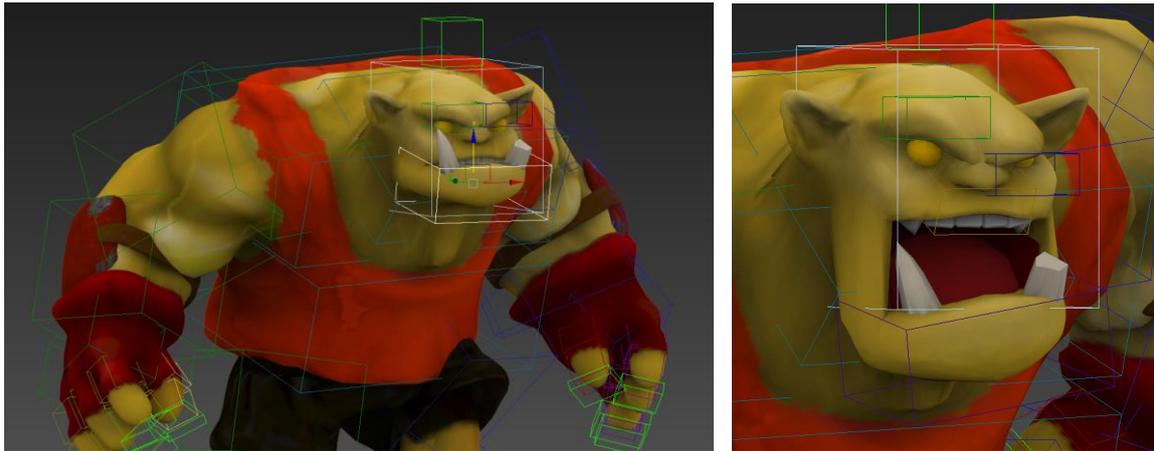


Figura 41: Sistema de huesos faciales aplicados al personaje.

Desde el botón *Schematic View* podemos visualizar a modo de árbol el esquema de dependencias del propio esqueleto con los nuevos huesos integrados y modificar las dependencias que existen entre ellos. Los huesos de la cara generados por nosotros aparecen marcados en rojo en la siguiente imagen:

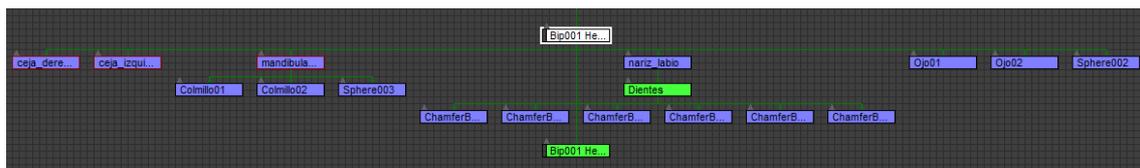


Figura 42: Vista de la opción Schematic View en 3D Studio Max.

Con el esqueleto en orden y colocado sobre nuestro personaje, llegó el momento de asignarle la malla del modelo al mismo. De esta manera, se vería afectada por el sistema de huesos y se deformaría con el movimiento esqueleto. Para ello le añadimos a la malla del modelo el modificador *Skin* de la lista de modificadores de *3D Studio*. Este modificador nos permite asignar los objetos que queremos que realicen la función de hueso, así que seleccionamos cada uno de los componentes del esqueleto que nos generó *Biped* y los que nosotros mismos creamos para la cara. Al instante, el modificador *Skin* ya realizaba su función, la malla del modelo ya se veía afectada por el movimiento del esqueleto y se deformaba en función del mismo, aunque no como esperábamos del todo.

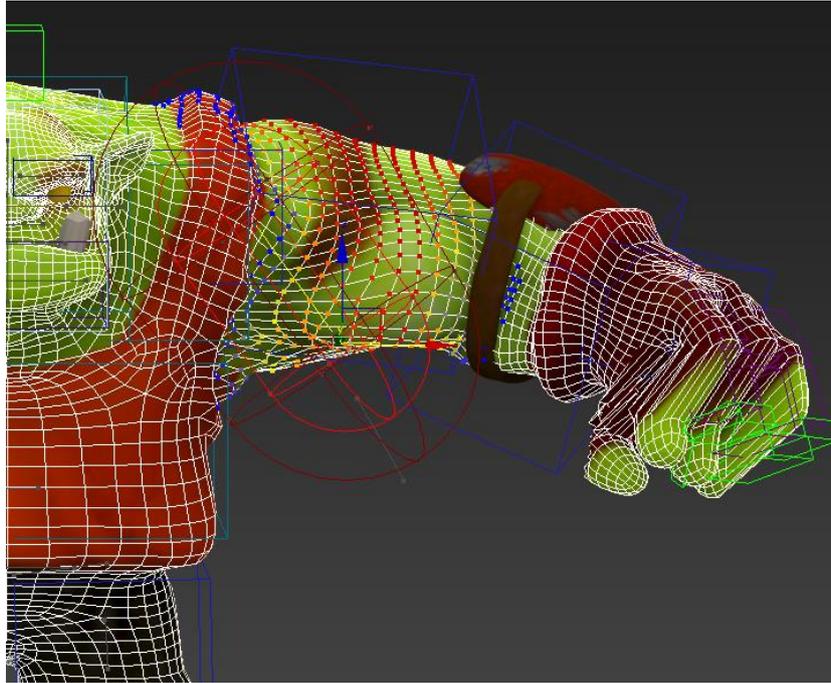


Figura 43: Vista de la opción *Envelope* de uno de los huesos del personaje.

¿Qué estaba ocurriendo?. Los huesos poseen una característica llamada *envelope* que se muestra gráficamente como una zona roja que determina la cantidad de vértices de la malla del modelo que se ven afectados por cada uno de los huesos. Estos vértices aparecen representados en varios colores según el peso o la fuerza que el hueso ejerce sobre éstos, siendo de color azul los vértices que menos fuerza de arrastre reciben y rojo los que más. De esta manera al flexionar una articulación, la geometría próxima a la misma se ve afectada por los dos huesos que la forman y la malla se dobla de manera natural.

Fijémonos en la mano izquierda de nuestro personaje, parte de los vértices que forman la mano no seguían a sus respectivos huesos. Esto ocurre porque al asignar el esqueleto al modelo, el programa realiza una aproximación de manera automática de cada zona de la malla a cada uno de los huesos adyacentes. La mano inicialmente estaba colocada cerca del hueso de la cadera y se veían influenciados por ese hueso. Problema que tuvimos que corregir de manera manual en las distintas zonas de la geometría del personaje.

Aquí comenzó una ardua tarea en la que tuvimos que revisar la malla entera en busca de imperfecciones. Para ello revisamos y ajustamos de manera gráfica las zonas de influencia *envelope* de cada hueso y ajustamos la fuerza ejercida sobre los vértices influenciados de cada uno de ellos. Una vez revisados todos, tuvimos que revisar y corregir paramétricamente los pesos de los vértices más problemáticos que se escaparon a la vista, esto lo conseguimos desde la tabla de pesos del modificador *Skin*.

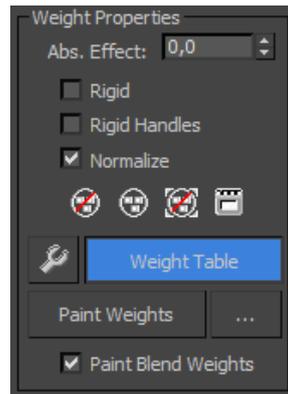


Figura 44: Vista de algunas de las opciones que ofrece el modificador *Skin*.

Skin Weight Table (Group1) — □

Edit Vertex Sets Options

Vertex ID	S	M	N	R	H	Bip001 L Foot	Bip001 L Toe0	nariz_labio	Bip001 R Thigh
#447	X	X	X			-	-	-	-
#448	X	X	X			-	-	-	-
#449	X	X	X			-	-	-	-
#450	X	X	X			-	-	-	-
#451	X	X	X			-	-	-	-
#499	X	X	X			-	-	-	-
#500	X	X	X			-	-	-	-
#501	X	X	X			-	-	-	-
#502	X	X	X			-	-	-	-
#503	X	X	X			-	-	-	-
#506	X	X	X			-	-	-	-
#507	X	X	X			-	-	-	-
#508	X	X	X			-	-	-	-
#509	X	X	X			-	-	-	-
#695	X	X	X			-	-	-	-
#696	X	X	X			-	-	-	-
#707	X	X	X			-	-	-	-
#714	X	X	X			-	-	-	-
#715	X	X	X			-	-	-	-
#717	X	X	X			-	-	-	-
#718	X	X	X			-	-	-	-
#719	X	X	X			-	-	-	-
#720	X	X	X			-	-	-	-
#721	X	X	X			-	-	-	-
#722	X	X	X			-	-	-	-

Figura 45: Vista de la table de pesos de cada uno de los vértices de la geometría del personaje.

En esta tabla, *3D Studio* nos proporciona un control completo de los pesos para cada vértice de la geometría con los huesos que poseen influencia directa sobre ellos. El peso aparece representado por un valor numérico decimal entre 0 y 1, donde 0 nos indica que no se ve afectado en absoluto y conforme más próximo a 1 es el valor, más relevancia tiene ese hueso sobre el vértice en concreto. El peso de cada vértice puede repartirse entre varios huesos, siendo 1 el mayor valor a repartir entre ellos. A continuación podemos ver peso de algunos de los vértices problemáticos de la punta de los dedos, donde observamos que, efectivamente, no todo el peso aparece sobre los huesos correspondientes:

Vertex ID	S	M	N	R	H	Bip001 L Forearm	Bip001 L Hand	Bip001 L Finger0	Bip001 L Finger01
#447	X	X	X			-	0,780	-	-
#448	X	X	X			-	0,172	0,608	-
#449	X	X	X			-	0,172	0,608	-
#450	X	X	X			-	0,735	-	-
#451	X	X	X			-	0,629	-	-

Figura 46: Vista de la table de los pesos de alguno de los vértices que forman la mano del personaje.

6.1.5 Animación

Antes de aventurarnos a animar sin más, realizamos un estudio sobre el tema y dimos con un documento *pdf*¹⁹ muy interesante llamado “the animators survival kit”[17] de *Richard Williams*, director de la película “¿Quién engañó a Roger Rabbit?”.

En él aparecen muy bien explicadas los conceptos y las bases de la animación y varios trucos y consejos para conseguir que las animaciones queden lo más vistosas posibles, lo cual, complementado con la gran cantidad de tutoriales que posee youtube nos permitió generar unas animaciones bastante aceptables dentro de nuestras limitaciones iniciales en este campo.

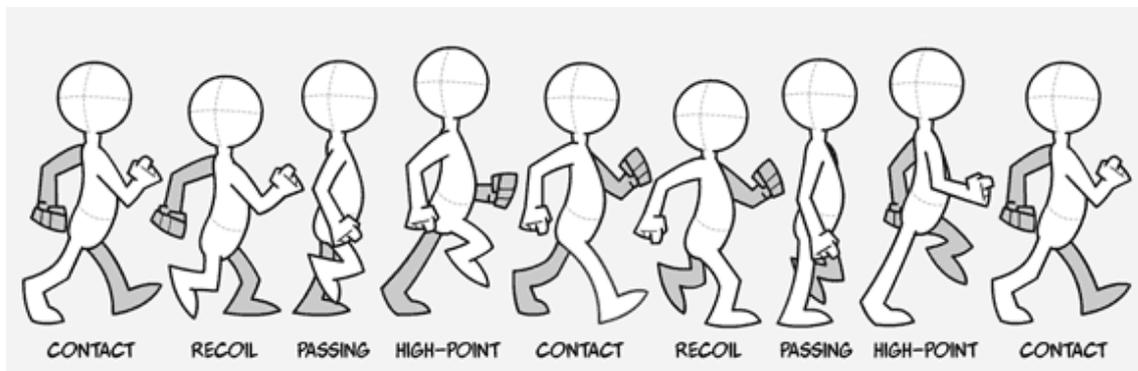


Figura 47: Fotogramas clave para el ciclo de una animación de andar.

En *Gnomore Gnomes* cada uno de los protagonistas cuenta con 25 animaciones propias diseñadas exclusivamente para cada uno de ellos. De esta forma pretendemos diferenciarlos no sólo por su aspecto, si no también en la manera en que se mueve cada uno de ellos, desde la forma de correr hasta en la manera en que cada protagonista realiza la celebración a la hora de anotar un tanto. Aún siendo distintas, las animaciones equivalentes en los distintos personajes poseen la misma duración entre ellas, de forma que, por ejemplo, la animación de correr para el *Orco* dura lo mismo que la del resto de personajes.

Las animaciones en *3D Studio* aparecen representadas como una barra de tiempo en la parte inferior de la ventana, dicha barra se puede configurar en función de si queremos que el tiempo aparezca representado en función de los *FPS* (frames por segundo) o en segundos. También nos permite indicar la cantidad de *FPS* que queremos que componga cada segundo, así como indicarle también la velocidad de reproducción de las animaciones. Podemos acceder a estas opciones desde el botón *Time configuration* situado en la esquina inferior derecha de la ventana principal de *3D Studio Max*.

¹⁹ http://www.upv.es/laboluz/tecnoimag/textos/the_animators_survival_kit_castellano.pdf

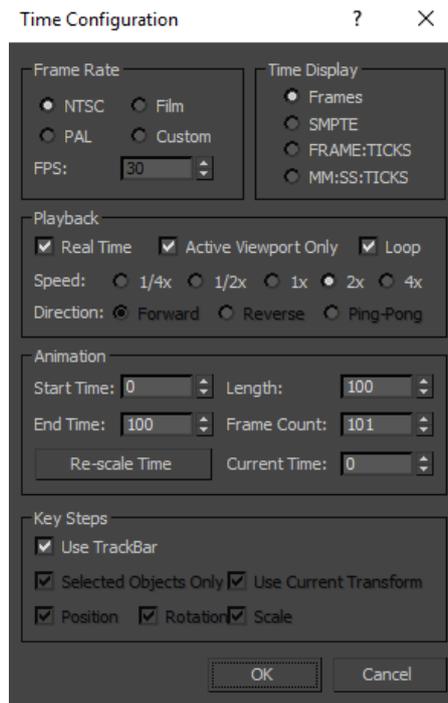


Figura 48: Vista de la ventana de configuración de tiempo de *3D Studio Max*.

Para cada uno de los elementos en pantalla, al seleccionarlos, el programa muestra unos puntos clave en los que genera de manera forma (aunque no siempre acertada acorde a nuestras preferencias) unas transiciones de movimiento entre ellos. De manera que si en el frame 30 (por ejemplo) tenemos un punto clave y movemos un objeto en el frame 40, el programa generará una transición entre ellos. Lo mismo ocurre con los huesos de nuestro personaje, al moverlos se generan puntos clave a modo de marcadores sobre la barra de tiempo como podemos ver en la imagen siguiente:



Figura 49: captura de la barra de tiempo de *3D Studio Max*.

Así que con cada uno de los huesos, en base a ciertos puntos clave cada cierto número de *frames*, (teniendo en cuenta que un segundo posee entre 25 y 30 según si elegimos PAL o NTCS) generábamos una postura acorde al movimiento que queríamos crear para que *3D Studio* generase las transiciones pertinentes. De esta manera obtuvimos una especie de esbozo de la animación la cual hubo que pulir para darle mayor naturalidad.

Para refinar las animaciones tuvimos que acceder internamente a ellas desde la herramienta *Curve editor*, situada en la barra superior de la interfaz de *3D Studio*. Desde esta opción tenemos control sobre la animación de cada objeto en escena (incluido los huesos) en sus 3 ejes XYZ, representados por distintos colores. Desde aquí, todo es completamente editable de forma paramétrica, lo que nos resultó esencial para algunas de las animaciones cíclicas como la de correr. Este tipo

de animaciones tienen la particularidad de acabar en la misma postura que empiezan. De esta manera conseguimos una animación que puede reproducirse en bucle durante todo el tiempo que deseemos.

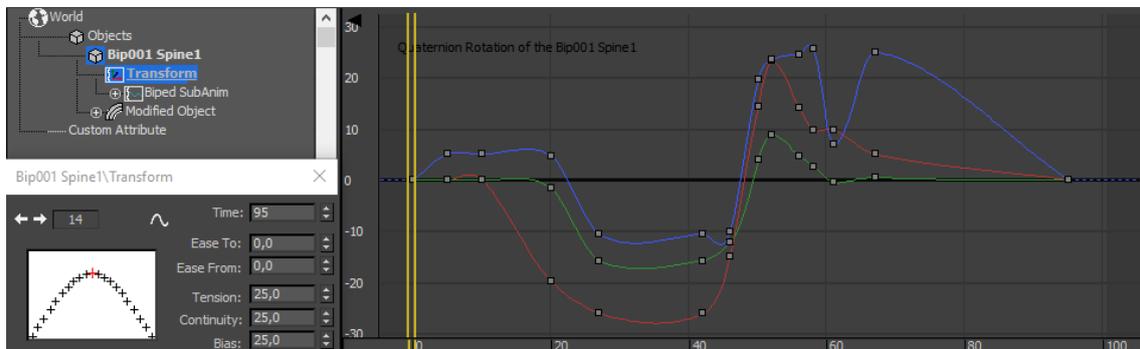


Figura 50: Vista de la ventana de curvas de animación de 3D Studio Max.

Pueden editarse multitud de parámetros que nos permiten conseguir dotar de mayor realismo las animaciones que fuimos generando. De esta forma simulamos la sensación del personaje del Orco al correr, de forma que diese la impresión de ser un movimiento más pesado que el de otros personajes debido a su constitución. A continuación pasamos a mostrar el esquema de las curvas de animación de todos los huesos que forman al Orco para su animación en ciclo de correr:

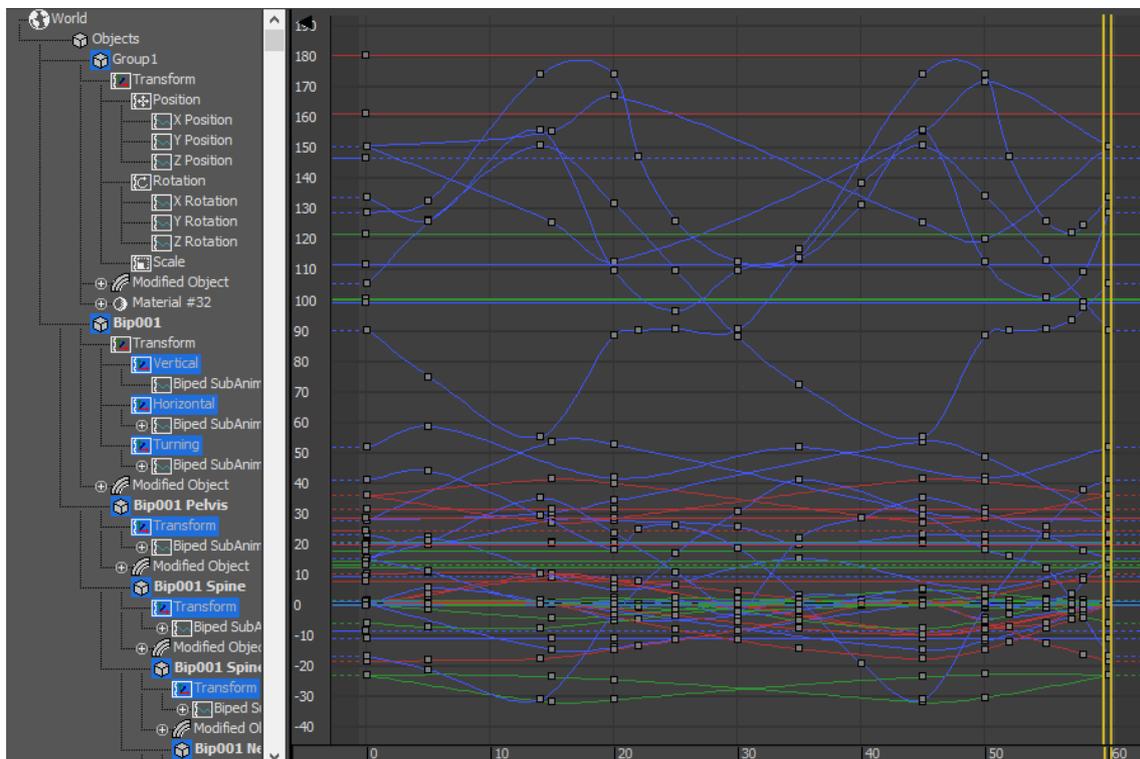


Figura 51: Vista de la ventana de curvas de animación generada para la animación de andar de uno de los personajes.

Una vez finalizada cada una de las animaciones que componen al orco, procedimos a exportarlas en formato FBX, ya que es un formato de fichero reconocible por *Unity 3D*. Una vez incluidas estas animaciones en el proyecto de *Unity*, el programador ya puede trabajar con ellas para integrarlas directamente en el videojuego.

6.2 Diseño de niveles

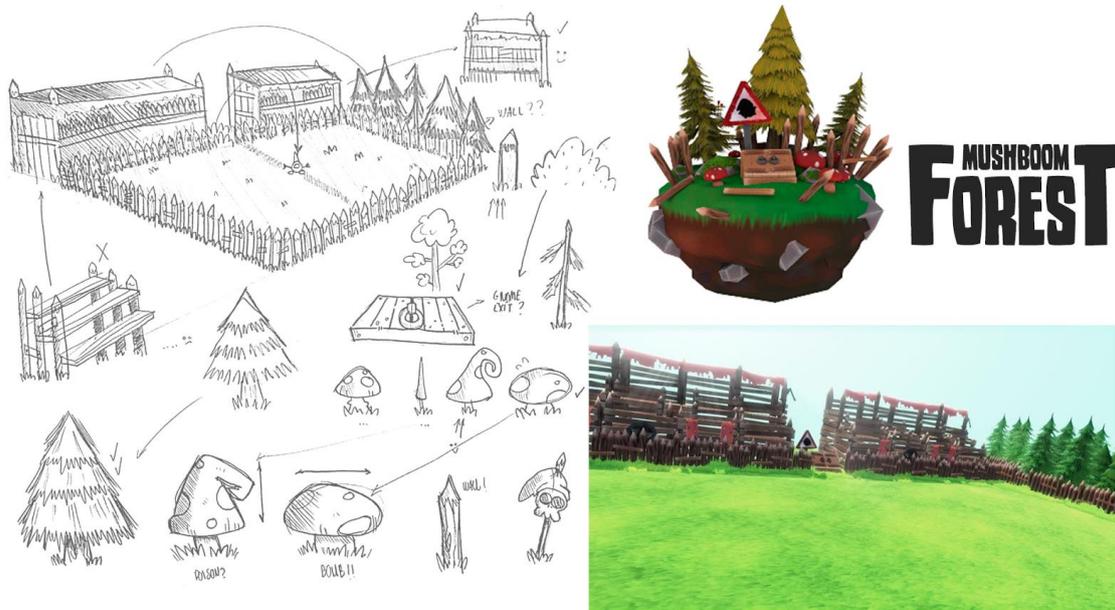


Figura 52: Imagen conceptual de uno de los escenarios, *Mushboom Forest*.

Por otra parte, el diseño de los distintos niveles que aparecen en *Gnomore Gnomes* fue algo menos complejo internamente que el de los personajes, aunque igualmente laborioso. También hubo una fase inicial donde desarrollamos varios bocetos, tanto de la vista global de cada escenario como de los elementos que aparecen en cada uno de ellos, hasta dar con los definitivos.

El proceso de modelado esta vez no pasó por *Zbrush*, puesto que los elementos que iban a aparecer en cada nivel tienen menos detalle que el que poseen los personajes. Todos ellos fueron generados a partir de geometría básica desde *3D Studio Max*. A continuación pasamos a explicar, a modo de ejemplo, el proceso de creación de uno de los elementos que aparecen en el escenario *Mushboom Forest*, en concreto uno de los árboles.

En primer lugar para crear el tronco, partimos de un objeto geométrico básico, en concreto un cono compuesto por seis polígonos de lado por cinco de alto como se muestra en la imagen.

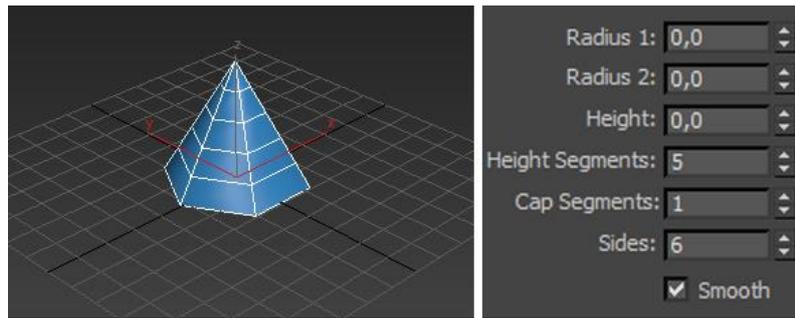


Figura 53: Captura de un modelo 3D generado con geometría básica desde *3D Studio Max*.

Una vez creada la base desde la cual partimos para modelar el tronco, pasamos a otorgarle la forma adecuada. Para ello disponemos de múltiples modificadores para la geometría que nos permiten, desde curvar la misma, hasta cubrirla de pelo. El modificador que utilizamos para dar forma al futuro tronco recibe el nombre de *edit Poly*. Este modificador es una herramienta que nos proporciona un control total para la geometría más interna del modelo 3D al que se lo aplicamos. De esta forma nos permite mover cada uno de sus vértices, aristas, cambiar el tamaño de los polígonos que lo forman, etc a nuestro antojo. Al ser un elemento del escenario y debido a que iban a aparecer multitud de ellos en escena, tampoco precisamos de un detalle demasiado grande. Para las hojas nos bastó con unos planos bastante simples, a los cuales también les aplicamos el modificador *edit Poly*, para darles la forma deseada y los repartimos haciendo copias de los mismos alrededor de la geometría del propio tronco. En la siguiente imagen podemos observar la evolución del árbol después de los pasos nombrados anteriormente.

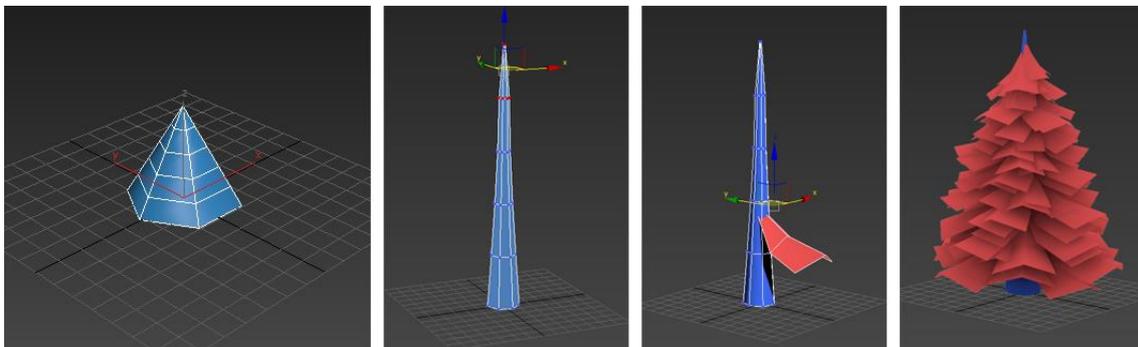


Figura 54: Evolución del modelo 3D del árbol.

El siguiente paso fue otorgarle una textura al objeto. Para ello primero obtuvimos el mapa de coordenadas del tronco y las hojas a través del modificador *UVW Map* y les dimos una aproximación de cono y de plano respectivamente. De esta manera la textura envuelve la malla de los objetos tratando a los mismos como geometrías básicas de ese tipo. Desde *Photoshop* creamos ambas texturas, con mención especial para la textura de las hojas, ya

que la imagen posee canal alpha, que es un tipo especial de imagen que tiene en cuenta las zonas que deseamos que sean transparentes. De esta manera con muy pocos polígonos (4 para el plano que conforma cada una de las hojas) conseguimos

un acabado bastante bueno y que representaba correctamente las ramas del modelo. A continuación mostramos el resultado final del árbol con sus texturas correspondientes:



Figura 55: Resultado del modelo final del árbol.

6.3 Introducción al sistema de partículas.

En un videojuego, la mayoría de los elementos que aparecen en un escenario son representados a través de modelos 3D (como los personajes, escenarios, etc) o imágenes planas como ocurre en el caso de los videojuegos en dos dimensiones. Son la forma más común de representarlos, sin embargo existen otros elementos menos tangibles a la hora de mostrar cómo los efectos de partículas que representan algunos elementos, como el fuego, destellos, agua, energía, etc. Estos pueden representarse desde Unity[18] gracias a la ayuda que nos aporta el propio *particle system* (sistema de partículas).

Las partículas pueden estar formadas tanto por modelos tridimensionales como por imágenes. Cada una de las partículas que conforman un sistema de partículas puede tener un comportamiento individual, pero cuando todas actúan juntas es cuando realmente dan forma al efecto que queremos conseguir, por ejemplo en el caso del fuego, que pasaremos a explicar posteriormente, existen partículas que pertenecen al humo y otras que pertenecen a la llama, pero una vez juntas es cuando dan el sentido total al efecto. Unity 3d también nos da la opción de aunar distintos sistemas de partículas y combinarlos entre sí.

Hay varios conceptos manejados por Unity 3D de forma paramétrica sobre los sistemas de partículas que nos pueden llegar a comprender su funcionamiento para usarlo en nuestro propósito y conseguir el efecto deseado, algunos de los más generales son:

- **Lifetime:** es el tiempo de *vida* que asignamos a cada una de las partículas del sistema de partículas. Con tiempo de vida nos referimos a la duración en la que cierta partícula se muestra en pantalla antes de desaparecer por completo. Este tiempo puede oscilar entre 2 números decimales de manera que podemos hacer que dentro del mismo sistema de partículas existan diferencias para cada una de ellas.

- **Emission Rate:** este concepto hace referencia a la cantidad de partículas emitidas por segundo teniendo en cuenta también el tiempo de vida de las mismas y el número máximo de ellas que queremos que se muestren por pantalla.
- **Shape:** Nos proporciona la opción de seleccionar la forma del objeto que representa al emisor. El emisor es un ayudante u objeto simbólico que no aparece en la escena pero delimita o da forma al emisor e indica la dirección en la que se proyectan las partículas, puede ser esférico, redondo, cuadrado, etc. También nos ofrece la opción de seleccionar desde qué parte de dicho ayudante queremos que emita las partículas como se generan.

Los conceptos nombrados anteriormente afectan directamente al comportamiento general del conjunto de partículas dentro del sistema, pero individualmente también podemos determinar que las partículas cambien durante el lapso de tiempo que están vivas. Cada una de las partículas de un sistema poseen un vector que determina la dirección y la fuerza con la que son emitidas en cada actualización de cuadro de la imagen. Este vector puede verse afectado, si lo deseamos, por algunas fuerzas externas tales como simulaciones de viento o gravedad. Tanto el tamaño, la rotación, el color, la transparencia o el movimiento de cada partícula puede variarse a lo largo de la vida de cada una de ellas de manera gradual o de forma brusca si así lo deseamos. A continuación hablaremos de cómo utilizamos el sistema de partículas para crear el fuego, el agua, explosiones, etc y veremos cómo utilizamos las diferentes opciones disponibles para la generación de los distintos efectos.

6.3.1 Ejemplo: Fuego

Gnomore Gnomes tiene varios efectos de partículas que refuerzan las distintas acciones o eventos que ocurren durante la partida, uno de estos efectos es el fuego.

El primer paso para crear el fuego fue observar el comportamiento del mismo, para ello visualizamos multitud de vídeos alojados en youtube de fuego real, animado, simulado, etc antes de empezar a desarrollar el nuestro. Llegamos a la conclusión de que deberíamos anidar dos sistemas de partículas para poder crearlo, puesto que por un lado tendríamos las partículas encargadas de representar las llamas y por otro las encargadas de mostrar el humo, pero ambas deberían actuar de manera simultánea. Para representar las partículas creamos la imagen encargada de mostrarlas. Para ello y teniendo en cuenta que queríamos que se solaparan entre ellas, de forma que pudiesen dar la sensación de formar un conjunto uniforme, otorgamos cierta transparencia a dicha imagen, dando como resultado el siguiente diseño:

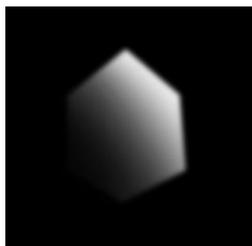


Figura 56: Imagen básica utilizada para representar las partículas.

Después de estudiar el comportamiento que deberían tener las llamas y diseñar la imagen que las representaría, decidimos que las partículas que la conforman debían un comportamiento en base al siguiente esquema::

- Las partículas deberían **emitirse de abajo hacia arriba**: para ello inicialmente enfocamos la dirección del emisor en esa dirección y le aplicamos cierta fuerza a cada una de las partículas desde la variable *start speed*, que es la encargada de otorgar la velocidad con la que se desplazan cada una de ellas. Sin embargo, el efecto no era del todo natural, ya que las partículas simplemente salían disparadas sin ningún tipo de aceleración. Así que decidimos enfocarlo de otra manera. Le dimos una velocidad más discreta, pero le asignamos un valor negativo a la gravedad, introduciendo dicho valor en la variable *Gravity modifier*. De esta manera obtuvimos un efecto de aceleración que quedaba bastante realista, pues al no haber gravedad, las partículas ascendían de la forma que buscábamos.

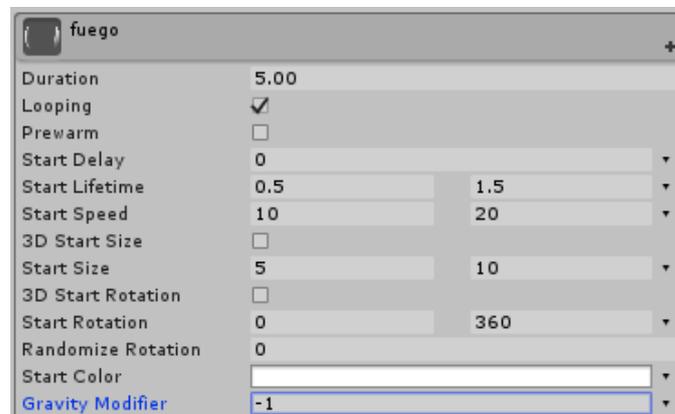


Figura 57: Vista de alguna de las opciones que nos ofrece el sistema de partículas de *Unity 3D*.

Como podemos observar hay otros valores que han sido modificados a parte de la gravedad. Algunos de ellos, como *Start Size*, se encargan de dar un tamaño aleatorio a las distintas partículas emitidas en un rango situado entre dos variables. Otro, como *Start Rotation*, en el que definimos la rotación inicial con la que aparece cada una de las partículas del sistema.

- Las partículas deberían **cambiar de color y acabar desapareciendo** a lo largo de su vida: El fuego real tiene variaciones de color dependiendo de la zona de la propia llama. Para ello debimos indicarle al sistema encargado de representar las llamas que las partículas emitidas se tintasen de las tonalidades que deseábamos a lo largo del tiempo de sus ciclos de vida. Para ello Unity dispone de una interfaz donde podemos indicar de forma gráfica los diferentes tonos que queremos que adquieran las partículas a lo largo del tiempo. Podemos situar marcadores a lo largo de la barra de color para indicar las transiciones de tintado e incluso el porcentaje de opacidad que queremos que adquieran.



Figura 58: Vista de la herramienta de tinte y opacidad de las partículas de Unity 3D.

- Las partículas deberían **desvanecerse** disminuyendo su tamaño a medida que avanza el tiempo: esto podemos indicárselo al sistema a través de una *curva bezier* totalmente configurable, donde el punto de inicio, que se sitúa en el nacimiento de cada una de las partículas del sistema, se encuentra en su valor máximo y el punto donde la partícula acaba desapareciendo por completo corresponde al valor mínimo asignable de la curva en cuestión. Unity 3D nos permite seleccionar entre varias curvas ya prediseñadas o generar la nuestra propia de manera totalmente personalizable para conseguir el efecto que buscamos.

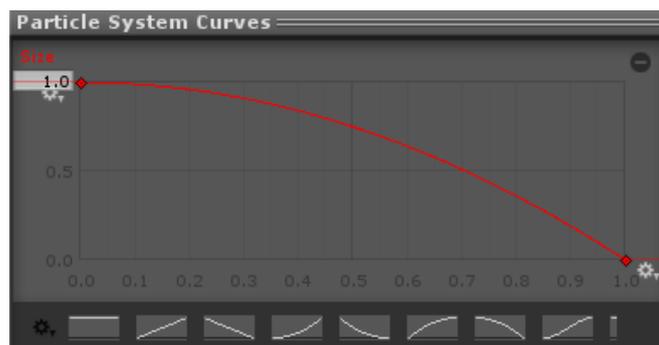


Figura 59: Vista de la curva Bezier encargada de las dimensiones de las partículas de Unity 3D.

Por otro lado, para el humo, creamos un nuevo sistema y lo anidamos con el de las llamas, de manera que ambos se vieran como un único sistema. El humo debía poseer un comportamiento similar al del fuego con algunas variaciones, por ejemplo el color de sus partículas sería en escala de grises, la vida de las partículas emitidas sería más larga y el tamaño de las mismas sería mayor.

Una vez aplicadas las configuraciones adecuadas sobre el sistema tanto de las llamas como el del humo, por fin obtuvimos el efecto que buscábamos, el cual podemos ver en la siguiente imagen:

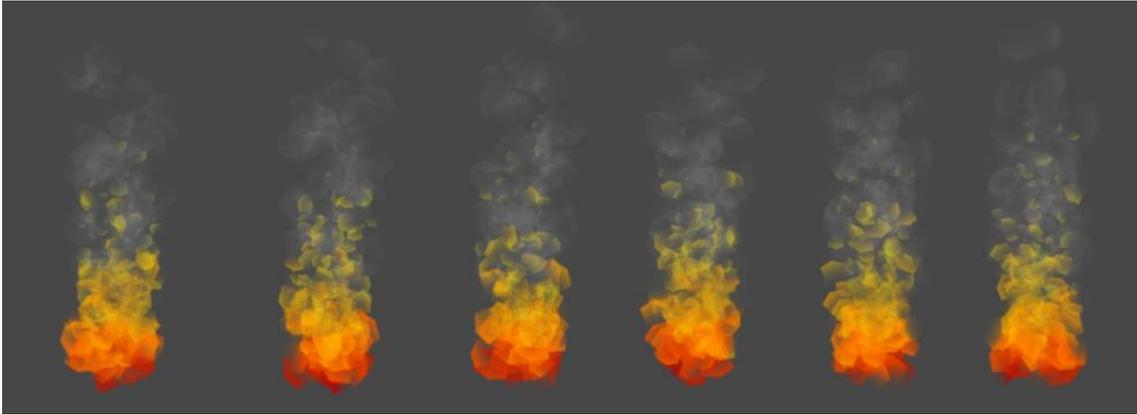


Figura 60: Muestra de algunos fotogramas de la animación del fuego.

En *Gnomore Gnomes*, aparecen muchos más efectos de partículas, la forma de conseguirlos consiste en la observación o el estudio previo de cómo se comporta en la vida real dicho efecto y una vez ha sido desglosado en varios puntos, con la ayuda de la gran cantidad de parámetros configurables que nos ofrece Unity, intentar imitarlos de la manera adecuada. A continuación mostramos algunos de los efectos de partículas que hemos creado desde cero para nuestro proyecto:

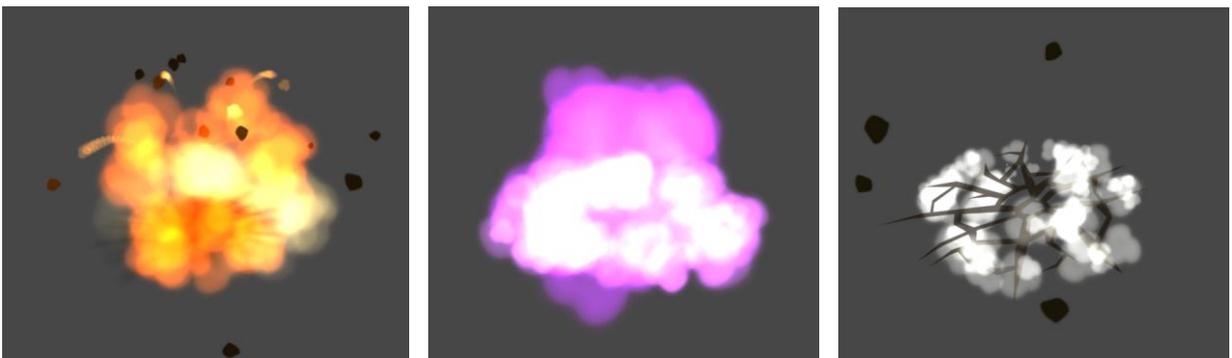


Figura 61: Muestra de algunos efectos de partículas que aparecen en el videojuego. De izquierda a derecha: Explosión, gas de una seta venenosa y fractura del terreno tras anotar un tanto.

7. Conclusiones y trabajo futuro

Han pasado casi dos años desde que tuvimos la idea inicial hasta el día de hoy donde tenemos una versión completamente jugable de nuestro propio videojuego. Poder dedicarnos a nuestra pasión y plasmarla en este proyecto ha sido un camino tan difícil como satisfactorio. Poder ver como unos simples bocetos en un trozo de papel han acabado transformándose en lo que hoy es *Gnomore Gnomes* ha sido como materializar un sueño. Aún así la pasión y la ilusión no garantizan que el producto vaya a tener éxito, y más teniendo en cuenta la cantidad de videojuegos que existen a día de hoy en el mercado.

Los primeros test no fueron todo lo bien que nos imaginamos y aunque llegamos incluso a plantearnos la posibilidad de abandonar, nuestra pasión nos ayudó a seguir avanzando y a superar los obstáculos que se interpusieron en el camino. A día de hoy podemos decir que estamos más que satisfechos de nuestro proyecto, supimos paliar los problemas que surgieron y gracias al esfuerzo pudimos corregirlos de la mejor manera posible siempre dentro de nuestras posibilidades. Durante el evento *Beer & Testing*, el cual hemos analizado en el segundo experimento, nos encontramos con muy buenos sentimientos al comprobar que varias personas totalmente desconocidas disfrutaron, compitieron y rieron con nuestro trabajo, haciendo que se levantaran de sus asientos. Fué un momento de gran orgullo ver la aceptación que obtuvimos y las felicitaciones de los jugadores que tuvieron la oportunidad de probarlo, los cuales llegaron a preguntarnos incluso “¿Cuándo sale a la venta?”.

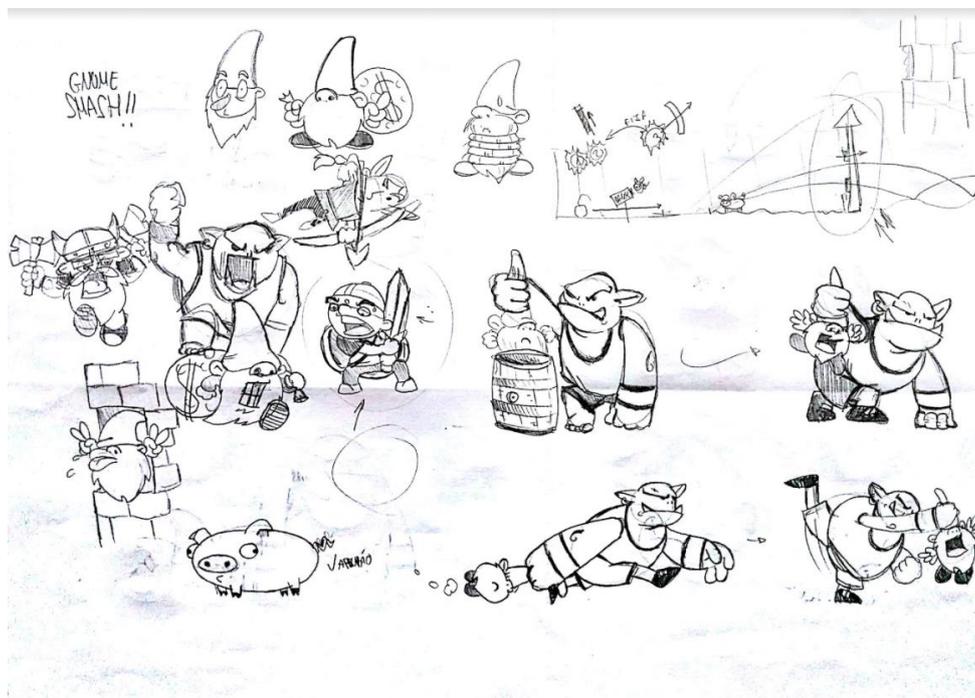


Figura 62: Bocetos iniciales para el videojuego *Gnomore Gnomes*.

A nivel personal la realización de este proyecto me ha sido muy útil para complementar mis estudios enfocándolos a una de mis grandes pasiones, ayudándome a organizar un proyecto desde cero y realizando un estudio de mercado antes de llevarlo a cabo. También me fue de gran ayuda a la hora de aprender a

utilizar nuevas tecnologías y herramientas tanto de diseño como de programación, que resultaron ser necesarias para poder realizar el producto.

Para el recuerdo queda una grabación de un momento del evento que nunca olvidaremos, y que, por ahora, no podemos dejar de ver²⁰. Nuestro objetivo estaba cumplido.

En cuanto al futuro de *Gnomore Gnomes*, la campaña de *crowdfunding* en cuya preparación nos hallamos inmersos mientras escribimos estas líneas, determinará si podremos seguir trabajando para mejorar y agrandar nuestro proyecto, y llegar así a cualquier jugador en el mundo que simplemente quiera divertirse con nuestra obra.

Nos gustaría concluir esta memoria con una cita del ya fallecido Satoru Iwata, antiguo director de Nintendo, un hombre que dedicó su vida a llevar la magia de los videojuegos a todos los jugadores del mundo, y que probablemente sea nuestro mayor referente:

“Una cosa que no ha cambiado -y no que cambiará- es nuestra propia naturaleza como una forma de entretenimiento. Como cualquier otro medio de entretenimiento, debemos crear una respuesta emocional para poder triunfar. Risas... miedo... gozo... rabia... afición... sorpresa... y sobre todo, orgullo por los logros conseguidos. Al final, generar esos sentimientos en nuestro jugadores es el verdadero juicio de nuestro trabajo. Esta es la forma definitiva de medir el éxito”.

- Satoru Iwata

²⁰ <https://www.youtube.com/watch?v=0sxpVJcD4e0>

Referencias y Bibliografía

[1] Información sobre la recaudación de ventas de videojuegos http://roibertjohan.blogspot.com.es/	8
[2] El videojuego <i>Bioshock 2</i> incorpora modo multijugador https://www.destructoid.com/2k-tells-us-why-bioshock-2-s-multiplayer-exists-162744.phtml	19
[3] El videojuego <i>Mass Effect 3</i> incorpora modo multijugador http://www.ign.com/articles/2011/10/10/mass-effect-3-multiplayer-confirmed	19
[4] El videojuego <i>Tomb Raider</i> incorpora modo multijugador http://www.ubergizmo.com/2013/01/tomb-raider-to-come-with-multiplayer-after-all/	19
[5] Las videoconsolas más vendidas https://en.wikipedia.org/wiki/List_of_best-selling_game_consoles	19
[6] Información sobre las ventas del videojuego <i>Minecraft</i> http://www.hobbyconsolas.com/noticias/minecraft-mas-100-millones-copias-vendidas-145454	21
[7] Información y datos sobre la comunidad de jugadores activa en Steam https://www.vidaextra.com/industria/steam-consigue-superar-los-125-millones-de-cuentas-activadas	20
[8] Información sobre la distribución de videojuegos en Steam en 2014 http://kotaku.com/nearly-40-of-all-steam-games-were-released-in-2016-1789535450	21
[9] Estadísticas de la plataforma Steam http://expandedramblings.com/index.php/steam-statistics/	21
[11] Top 50 juegos durante el año 2014 http://www.eurogamer.net/articles/2015-01-02-readers-top-50-games-of-2014	23
[12] Nintendo y el contenido extra en sus videojuegos http://es.ign.com/mario-kart-8-wii-u/80498/feature/nintendo-y-los-dlc	24
[13] Documentación Zbrush http://docs.pixologic.com/user-guide/customizing-zbrush/zscripting/	51
[14] Documentación 3D Studio Max.....	52
Andrew Gahan “3ds Max Modeling for Games: Insider's Guide to Game Character, Vehicle, and Environment Modeling”	
[15] Documentación Adobe http://tv.adobe.com/es/show/tutoriales-para-disenadores/	54
[16] Información sobre el concepto de rigging http://www.ite.educacion.es/formacion/materiales/181/cd/m10/rigging.html	63
[17] Richard Williams “Kit de supervivencia del animador” http://www.upv.es/laboluz/tecnoimag/textos/the_animators_survival_kit_castellano.pdf	68

[18] Documentación Unity 3D https://docs.unity3d.com/Manual/index.html	73
[15] Vídeo de los usuarios probando <i>Gnomore Gnomes</i> durante el evento B&T https://www.youtube.com/watch?v=0sxpVJcD4e0	79

Material y transparencias proporcionadas por cursos realizados en centro de formación de la UPV:

- Curso de animación de personajes de videojuegos.
- Curso 3D Studio Max (Autodesk).
- Introducción al desarrollo de videojuegos en Unity 3D.

Índice de Ilustraciones

Figura 1: Logo del videojuego	8
Figura 2: Captura de una funcionalidad con la partida final.	15
Figura 3: partida multijugador local de Halo: Combat Evolved	17
Figura 4: Videoconsola Nintendo Wii	18
Figura 6: Captura de la interfaz principal de Steam.....	20
Figura 8: Captura de la interfaz principal de Xbox Live	21
Figura 9: Capturas de dos partidas en dos escenarios distintos de Towerfall	22
Figura 10: Capturas de dos partidas en dos escenarios distintos de Overcooked.....	23
Figura 11: Capturas de dos partidas en dos escenarios distintos de Mario Strikers: Charged	25
Figura 12: Capturas de dos partidas en dos escenarios distintos de Stikbold!.....	27
Figura 13: Mapa de características de Gnomore Gnomes	36
Figura 14: Captura de la ficha perteneciente a la tarea de desarrollo del personaje	38
Figura 15: Captura de la ficha perteneciente a la tarea de desarrollo del personaje	39
Figura 16: Captura del tablero de tareas del 19 de Diciembre.....	39
Figura 17: Captura de una de las animaciones del personaje Cabra	40
Figura 18: Captura de una de las partidas que tuvo lugar durante el primer experimento.....	45
Figura 19: Resultados de la pregunta 1.....	46
Figura 20: Resultados de la pregunta 2	47
Figura 21: Resultado de la pregunta 3.....	48
Figura 22: Resultado de la pregunta 4	48
Figura 23: Resultado de la pregunta 5	49
Figura 24: Resultado de la pregunta 6	50
Figura 25: Resultado de la pregunta 7.....	51
Figura 30: Bocetos iniciales de los personajes	56
Figura 31: Evolución conceptual de uno de los personajes.	57
Figura 32: Proceso inicial de modelado 3D del personaje.	58
Figura 33: Vista de la herramienta Zspheres en la interfaz de Zbrush.....	58
Figura 34: Generación de la estructura del personaje con la herramienta Zspheres.....	59
Figura 35: Vista de la herramienta Adaptive Skin en la interfaz de Zbrush.	59
Figura 36: Modelo 3D resultante al aplicar Adaptive Skin.....	60
Figura 37: Modelo 3D en la fase final de modelado desde Zbrush.	60
Figura 38: Evolución gráfica del personaje Lara Croft de la saga Tomb Raider.	61
Figura 39: Vista de la herramienta Decimation Master del programa Zbrush.....	61
Figura 40: Imagen comparative del modelo 3D con distinta carga polygonal.	62
Figura 41: Obtención del mapa de coordenadas desde Zbrush.	63
Figura 42: Vista de la herramienta exportar textura de Zbrush.	63
Figura 43: Esquema de las capas que forman la textura en Photoshop.	64
Figura 44: Resultado de la textura tras el retoque aplicado en Adobe Photoshop.	64
Figura 45: Sistema de huesos generados desde 3D Studio Max.	65
Figura 46: Vista del esqueleto de Biped en 3D Studio Max.	65
Figura 47: Sistema de huesos faciales aplicados al personaje.	66
Figura 48: Vista de la opción Schematic View en 3D Studio Max.	66
Figura 49: Vista de la opción Envelope de uno de los huesos del personaje.....	67
Figura 50: Vista de algunas de las opciones que ofrece el modificador Skin.	68
Figura 51: Vista de la table de pesos de cada uno de los vértices de la geometría del personaje.	68
Figura 52: Vista de la table de los pesos de alguno de los vértices que forman la mano del personaje.....	68
Figura 53: Fotogramas clave para el ciclo de una animación de andar.	69
Figura 54: Vista de la ventana de configuración de tiempo de 3D Studio Max.	70

<i>Figura 55: captura de la barra de tiempo de 3D Studio Max.</i>	<i>70</i>
<i>Figura 56: Vista de la ventana de curvas de animación de 3D Studio Max.</i>	<i>71</i>
<i>Figura 57: Vista de la ventana de curvas de animación generada para la animación de andar de uno de los personajes.</i>	<i>71</i>
<i>Figura 58: Imagen conceptual de uno de los escenarios, Mushboom Forest.</i>	<i>72</i>
<i>Figura 59: Captura de un modelo 3D generado con geometría básica desde 3D Studio Max.</i>	<i>73</i>
<i>Figura 60: Evolución del modelo 3D del árbol.</i>	<i>73</i>
<i>Figura 61: Resultado del modelo final del árbol.</i>	<i>74</i>
<i>Figura 62: Imagen básica utilizada para representar las partículas.</i>	<i>75</i>
<i>Figura 63: Vista de alguna de las opciones que nos ofrece el sistema de partículas de Unity 3D.</i>	<i>76</i>
<i>Figura 64: Vista de la herramienta de tintado y opacidad de las partículas de Unity 3D.</i>	<i>77</i>
<i>Figura 65: Vista de la curva Bezier encargada de las dimensiones de las partículas de Unity 3D.</i>	<i>77</i>
<i>Figura 66: Muestra de algunos fotogramas de la animación del fuego.</i>	<i>78</i>
<i>Figura 67: Muestra de algunos efectos de partículas que aparecen en el videojuego. De izquierda a derecha: Explosión, gas de una seta venenosa y fractura del terreno tras anotar un tanto.</i>	<i>78</i>
<i>Figura 68: Bocetos iniciales para el videojuego Gnomore Gnomes.</i>	<i>79</i>

Anexo:

Flujo de una partida en *Gnomore Gnomes*

A continuación vamos a describir el flujo de una partida del videojuego *Gnomore Gnomes* en su última versión mínima jugable hasta la fecha. Mostramos esquema y una breve explicación del contenido de cada una de las distintas vistas del videojuego y de las transiciones que existen entre ellas:

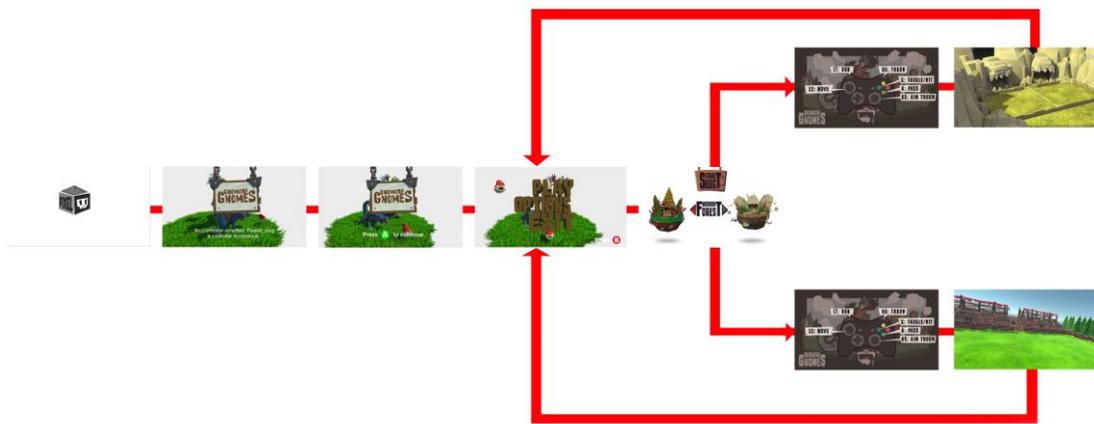


Figura Anexo 1: Vista esquemática del flujo de ejecución de *Gnomore Gnomes*.

Como podemos ver en la imagen anterior, mostramos el flujo de ejecución normal para la versión actual del videojuego *Gnomore Gnomes*, el cual pasamos a detallar a continuación en cada una de las diferentes vistas que ofrece.



Figura Anexo 2: Vista del logo de *Dead Pixel*.

Se muestra durante unos instantes el logo de *Dead Pixel*. Después de unos segundos el logo desaparece de la pantalla dando paso al inicio del juego.

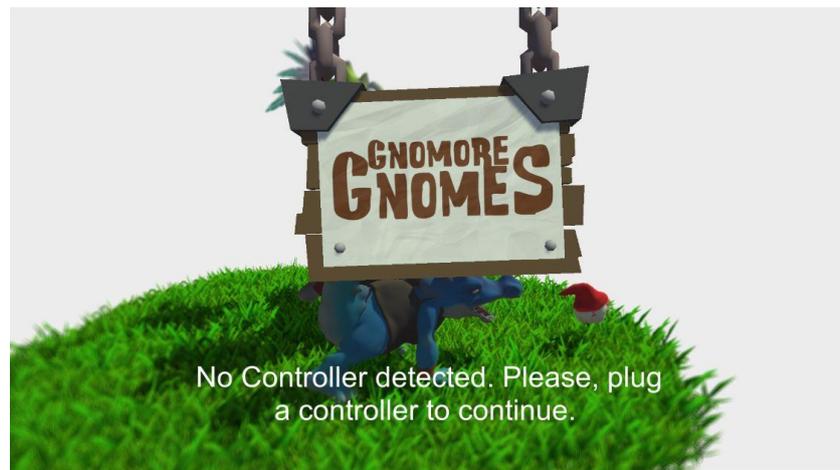


Figura Anexo 3: Pantalla de detección de controlador de juego.

Se muestra por pantalla la vista de la imagen anterior. Esta vista se mantiene a la espera hasta que un usuario conecta un controlador USB para juegos en uno de los puertos del ordenador.

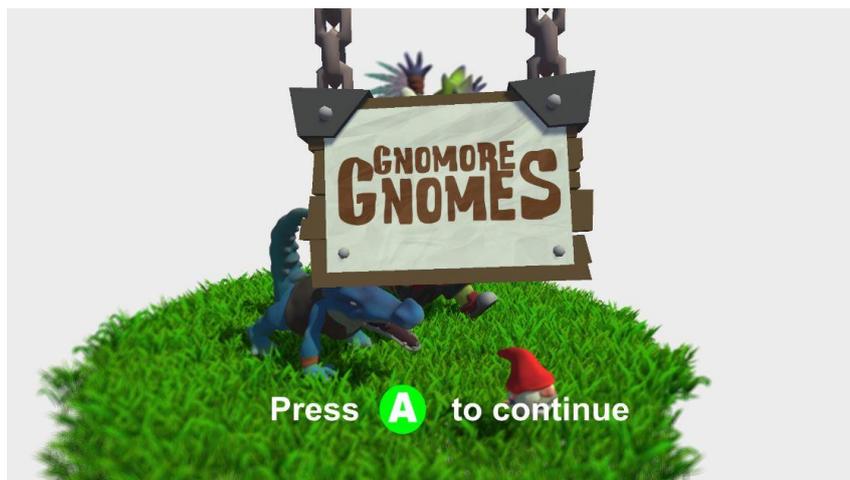


Figura Anexo 3: Pantalla de confirmación de inicio del videojuego.

Una vez conectado un controlador, se pide confirmación al usuario para iniciar el videojuego pidiéndole que pulse el botón "A" o *botón 1* del mando.



Figura Anexo 4: Vista del menú de inicio.

La imagen anterior pertenece a la vista del menú de inicio. Aquí se muestran las diferentes opciones que se le ofrecen al usuario.

- Play: Al pulsar esta opción el usuario podrá elegir el escenario donde desea iniciar la partida para empezar la partida.
- Options: Aparecen las diferentes opciones que permiten al usuario configurar el videojuego asu gusto. El menú de opciones se encuentra actualmente en desarrollo, por lo que no es posible mostrar una imagen de dicha vista.
- Exit: El juego se cierra completamente, dando paso al escritorio de Windows.



Figura Anexo 5: Pantalla de selección de escenario.

Una vez el usuario ha pulsado la opción *Play* se mostrará una vista donde se le dará a elegir a los jugadores el escenario en el que quieren realizar la partida, podrán elegir entre dos por el momento, que son los que están disponibles en la versión actual del videojuego. Al pulsar el botón "A" comenzará la carga de la partida en el escenario previamente seleccionado.



Figura Anexo 6: Pantalla de carga del juego.

Durante el tiempo de carga, se muestra la imagen anterior de modo que el usuario pueda familiarizarse con los controles del videojuego antes de iniciar la partida. Una vez haya cargado esta vista se mantendrá durante el tiempo deseado hasta que uno de los usuarios pulse el botón A. A continuación se dará paso al inicio de la partida.



Figura Anexo 7: Vista del escenario *Rock'n Troll Hills*.

En la imagen se muestra *Rock'n Troll Hills*, nombre que recibe el escenario y donde se efectuará la partida si el usuario seleccionó .



Figura Anexo 7: Vista del escenario *MushBoom Forest*.

En la imagen anterior se muestra *MushBoom Forest*, nombre que recibe este escenario y donde se efectuará la partida si el usuario seleccionó.

Una vez finalizada la partida el videojuego vuelve a la pantalla del menú principal y el flujo de ejecución continúa desde ese punto hasta que el usuario decide dejar de jugar y selecciona la opción *Exit* de dicho menú.