Universidad
Rey Juan Carlos

Escuela Técnica Superior de Ingeniería Informática

Grado en Diseño y Desarrollo de Videojuegos

Curso Académico 2020/2021

Trabajo Fin de Grado

Videojuego basado en creación, edición y subasta de estructuras en Unreal Engine

Home Factory

Autor: Darío González Fernández

Tutora: María Zapata Cáceres

Mayo 2021


BY NC SA

Agradecimientos

Con todo el cariño:

A todas aquellas personas que han colaborado y formado parte de este proyecto.

A aquellos que me han apoyado llueva, nieve o haga sol ***incondicionalmente****.*

A todos aquellos a los que mi experiencia pueda servir de inspiración y motivación.

A todos los que habéis hecho que hoy Home Factory sea una realidad.

Resumen

Actualmente nos encontramos ante un mercado donde prácticamente todos los diversos géneros de videojuegos están altamente explotados. Ya no resulta suficiente destacar únicamente con las ventajas y beneficios de un videojuego, por lo que llegamos a una necesidad de buscar alternativas para crear expectativas e interés por el juego.

En este trabajo, se plantea el desarrollo de un prototipo de videojuego en tercera persona: Home Factory, desarrollado en Unreal Engine y con estética Cel Shading y uso de partículas como efectos añadidos, que, como innovación, combina elementos de construcción, personalización y subastas en un único título tratando de explotar al máximo la creatividad del jugador mediante la creación de estructuras y la sensación de progreso gracias a la incorporación de un sistema de subastas en la que se valora la construcción del jugador. Mediante la correcta combinación de dichos elementos, se propone un juego sin objetivos ni tiempos donde el jugador es el único capaz de establecer estos puntos.

En este documento se realiza una investigación de juegos a lo largo de la historia cuya temática era la construcción, incorporaban elementos similares o títulos que incorporaban un sistema de subastas. Basado en las conclusiones extraídas de la investigación, se ha realizado el diseño y creación de un prototipo para Home Factory. Posteriormente, se ha realizado una validación del prototipo, en la que se evalúa si las mecánicas de juego que combinan elementos de construcción, personalización y subastas pueden ser atractivas para los jugadores.

Palabras Clave:

- Creación
- Estructuras
- Cel Shading
- Partículas
- Subasta
- Unreal Engine
- Videojuego

Abstract

We are currently facing a market where almost all different genres of video games are highly exploited. It is no longer enough to stand out only with the advantages and benefits of a video game, so we came to a need to find alternatives to create expectations and interest in the game.

In this project, we propose the development of a prototype of a third-person video game: Home Factory, developed in Unreal Engine, with Cel Shading aesthetics and the usage of particles as added effects, which, as an innovation, combines elements of construction, customization and auctions in a single title trying to exploit the maximum creativity of the player through the creation of structures and the feeling of progress thanks to the incorporation of an auction system in which the player's construction is valued. Through the correct combination of these elements, a game without objectives or times is proposed where the player is the only one capable of establishing these points.

In this document an investigation is made of games throughout history whose theme was construction, incorporated similar elements or titles that incorporated an auction system. Based on the conclusions drawn from the research, the design and creation of a prototype for Home Factory was carried out. Subsequently, a validation of the prototype has been carried out, evaluating whether game mechanics that combine elements of construction, customization and auctions can be attractive to players.

Keywords:

- Creation
- Structures
- Cel Shading
- Particles
- Auction
- Unreal Engine
- Videogame

Contenido

Índice de ilustraciones

Glosario

RPG	**Role-Playing Game**
Unreal Engine	Motor de videojuegos creado por Epic Games aparecido inicialmente con el Shooter Unreal en 1998. Basado en código C++ y facilitando la programación de videojuegos mediante un sistema de BluePrints
Cel Shading	Conocido también como "Sombreado Plano" o "Toon Shading" es un tipo de renderización no fotorrealista diseñado para que los gráficos generados por computación parezcan dibujados a mano mediante un sombreado plano, imitando el estilo de comics o dibujos animados
SandBox	Videojuego no lineal que ofrece al jugador la posibilidad de decider cómo continuar, cómo recrear su propio mundo, asumiendo las consecuencias de sus acciones. No hay un objetivo definido
ARPG	Action Role-Playing Game
MMORPG	Massively Multiplayer Online Role-Playing Game
MOBA	Multiplayer Online Battle Arena
RTS	Real Time Strategy
NES	Nintendo Entertainment System
Steam	Plataforma de videojuegos desarrollada por Valve
Shoot 'em up	Género de videojuegos donde el jugador controla un objeto, ya sea un vehículo o un personaje, que dispara contra constantes oleadas de enemigos.
Farmeo	Acto repetitivo de matar enemigos cuya finalidad se basa en obtener oro, experiencia, puntos de facción u objetos.
Looteo	Acto que se centra en la recolección de objetos o la moneda en cuestión del propio juego. La forma más habitual de looteo suele ser abrir cofres o cajas que encontremos por el camino o recolectar ítems que suelten los enemigos.
NPCs	Non Playable Characters

DLC	DownLoadable Content o contenido descargable. Es un contenido que no está incluido en el juego base original y puede obtenerse por separado o que estando incluido se desbloquea mediante la compra de contenido adicional.
Snap-In	Función de auto-ajuste que encaja las piezas unas con otras automáticamente
Override	Sobreescritura sobre una función que ya está definida en la clase padre
Clipping	Error gráfico que sucede cuando dos texturas están a la misma altura
Pivot	Punto de referencia sobre el que se mueve, rota o escala una figura
Popping	Efecto visual no deseado por el que los objetos aparecen en pantalla de manera espontánea según nos vamos acercando a ellos dando la sensación de que se van creando a nuestro paso.

Capítulo 1 - Introducción

En este capítulo se hablará sobre la historia de los videojuegos que incorporan elementos principales en su jugabilidad como la posibilidad de edición/creación de niveles, creación de estructuras o sistemas de subasta y compraventa.

Se dedicará una pequeña sección a hablar de la posición de Home Factory, entre otros títulos actuales del mercado.

1. Historia

Todos los videojuegos de los que disfrutamos actualmente incorporan niveles que han sido previamente diseñados siguiendo una estructura y una lógica determinada en función del uso que se dé al tipo de nivel en concreto. La diversidad a la hora de crear los niveles dependerá del género (estrategia, acción, sandbox, RPG, ARPG, MMORPG, RTS, conducción, MOBA, arcade, deportes, etc.) y de la modalidad (un solo jugador, multijugador, juego abierto, juego lineal) del juego en cuestión.

Sin embargo, a lo largo de la historia de los videojuegos, muchos títulos han incorporado la libertad de poder diseñar niveles dentro del propio juego a sus usuarios.

A principios de **1980**, la empresa Namco [1] lanza al mercado distribuido de la mano de Midway Games, **Pac-Man** [2]. Desde su lanzamiento, el videojuego ha sido un fenómeno mundial en la industria de los videojuegos, llegando a tener el récord Guiness del videojuego arcade más exitoso de todos los tiempos, destronando al ya famoso **Space Invaders** [3]. Un año después, Ed Averett diseña y programa K.C. Munchkin! ó **Munchkin** [4] como llegó a Europa.

Munchkin partía de la misma base de **Pac Man**, sin llegar a ser un clon directo e incorporando algunas diferencias entre ambos juegos, entre ellas y la que nos resultará más destacable en nuestra historia temporal, el modo programable. Además, **Munchkin** incorporaba lo que podríamos considerar uno de los primeros

diseñadores de niveles de la historia en su modo programable, permitiendo al jugador poder diseñar sus propios laberintos.

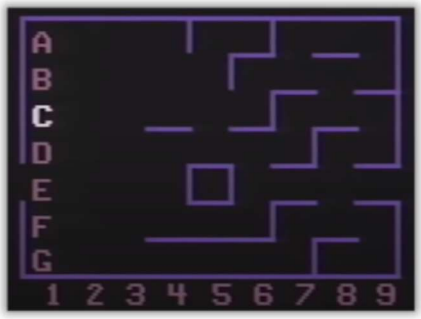


Ilustración 1 - Modo programable Munchkin

En **1983**, *Brøderbund* [5] publica **Lode Runner** [6] incorporando su editor de niveles y haciendo que la popularidad del juego aumentara hasta el punto de que revistas del momento como *Computer Gaming World* organizaran concursos premiando al que crease el mejor nivel. El editor de niveles de **Lode Runner** se convirtió en algo totalmente revolucionario, no sólo por su simplicidad de uso, sino porque debido a la jugabilidad, animó a muchos jugadores a dejar de lado los niveles predefinidos que incorporaba el propio juego a convertirse en "desarrolladores". Asimismo, todos aquellos niveles que fuesen enviados directamente a Brøderbund, podrían ser considerados para incorporación en la secuela del mismo título, **Championship Lode Runner**, un videojuego que seguía las mismas directrices de su predecesor, pero incluyendo niveles mucho más desafiantes para el jugador.

Ilustración 2 - Menú principal de Lode Runner

Ilustración 3 - Portada Championship Lode Runner

En **1984** se lanza en Japón **Excitebike** [7], juego que debutó en el lanzamiento para la consola Famicom [8] en ese mismo año que consistía en correr sólo, contra motoristas manejados por la consola y contra límites de tiempo. Al igual que sucedió con **Lode Runner**, impulsó a incontables jugadores hacia el mundo de la creación y el diseño de videojuegos.

Ilustración 4 - Menú principal Excitebike y su editor

En **1984** se lanza también el primer videojuego diseñado por Will Wright, publicado por Brøderbund , **Raid on Bungeling Bay** [9]que incorporaba una característica de evolución de fábricas y sirvió de inspiración para Will Wright, ya que disfrutaba más de crear los mapas para el videojuego que jugar el propio juego, dando origen en **1989** al primer título de la saga **SimCity** [10].

Ilustración 5 - SimCity

SimCity es un título basado en la creación de ciudades para simular la construcción y desarrollo de una ciudad con un amplio sentido de urbanismo, dando lugar a un nuevo género dentro del mundo de los videojuegos.

En **1992** se lanza al mercado desarrollado por Gremlin Interactive [11] y publicado por *Virgin Interactive* [12], **Greg Norman's Golf Power** [13] para NES (nombre por el que se conocía a la versión de la consola FAMICOM en el mercado europeo y americano). Aparte de llevar el nombre y la imagen del jugador australiano, fue distinguido de la mayoría de los videojuegos de golf por incorporar un modo de diseño de niveles, donde el jugador podía crear y personalizar su propio campo de 18 hoyos.

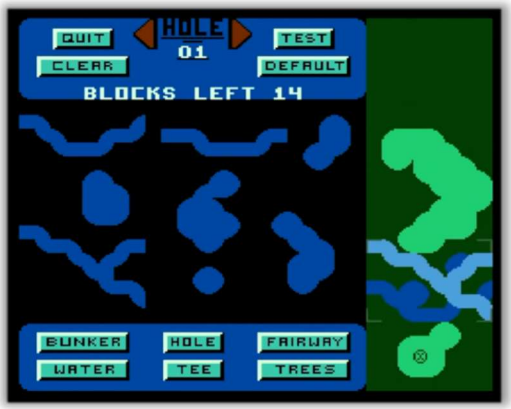


Ilustración 6 - Editor Greg Norman's Golf Power

Pero no todos los videojuegos incorporaban de forma nativa o en sus opciones herramientas de diseño de niveles, en mayo de ese mismo año aparece **Wolfenstein 3D** [14]y en el mismo mes de lanzamiento, Bill Kirby lanza una herramienta de creación y edición de niveles, **MapEdit** [15]. **Wolfenstein 3D** sirvió de fuente de

inspiración para el género de los shooter en primera persona dando lugar a títulos tan revolucionarios como **Doom** [16] (1993) o **Quake** [17] (1996).

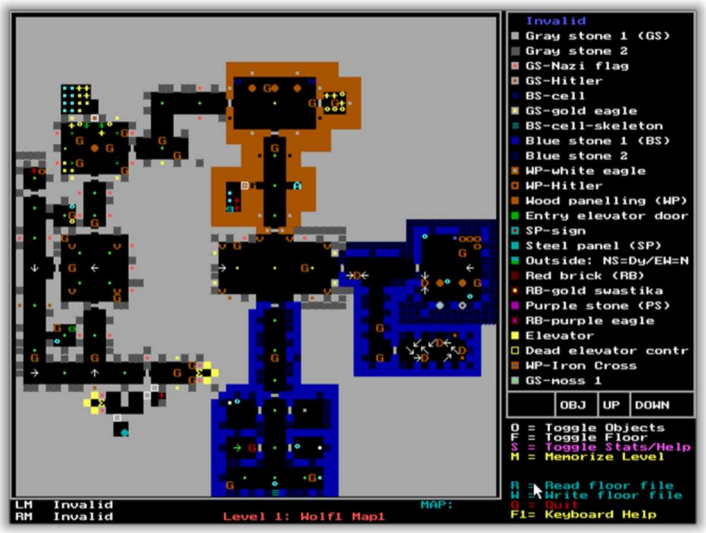


Ilustración 7 - Herramienta de creación y edición de niveles MapEdit

Ilustración 8 - Portadas de los principales shooter en primera persona del momento

Raphaël Quinet, Brendon Wyber y Renaud Paquay diseñaron la primera herramienta de creación y edición para Doom un año después de su lanzamiento, en **1994**, **Doom Editing Utilities** [18] (también conocido como DEU). Posteriormente, John Romero y John Carmack lanzaron el primer editor de niveles para **Quake** en **1996**, **QuakeEd** [19].

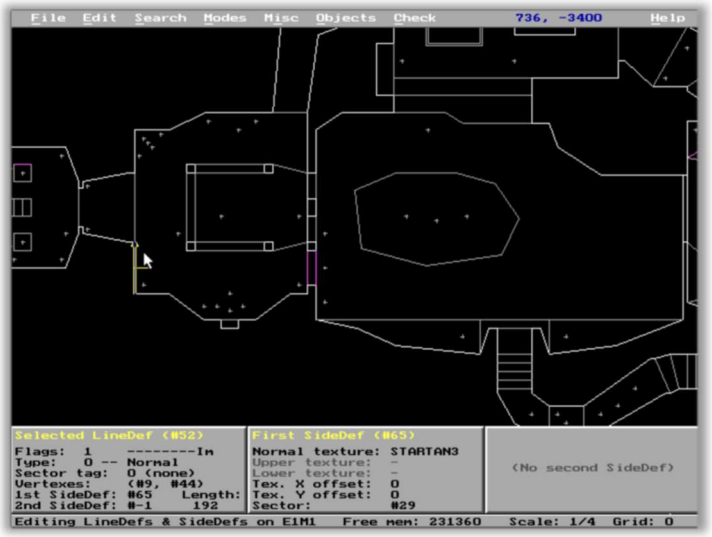


Ilustración 9 - Doom Editing Utilities

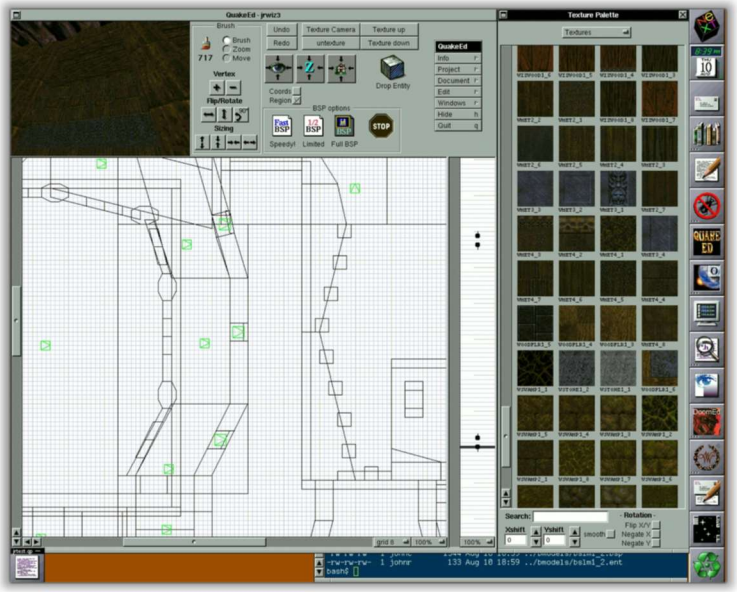


Ilustración 10 - QuakeEd

En línea de lo ya mencionado acerca de **SimCity** (1989), juegos basados directamente en creación y gestión, aparece en **1999 RollerCoaster Tycoon** [20]. **Este** título nos permite crear y gestionar un parque de atracciones por completo desde cero y que al igual que su fuente de inspiración, dará lugar a una saga de títulos.

Ilustración 11 - RollerCoaster Tycoon

Aunque el género de creación y edición de niveles seguía evolucionando y se empezaban a generar herramientas externas a los propios videojuegos para la creación de niveles, la inclusión de editores de nivel dentro de los propios videojuegos no dejó de existir.

En septiembre del año **2000** se lanzó la secuela del videojuego del conocido patinador estadounidense Anthony Frank Hawk más conocido como Tony Hawk, **Tony Hawk's Pro Skater 2** [21]. La principal novedad de este título no residía únicamente en los niveles nuevos y la inclusión de todos los niveles de la primera entrega, sino que incorporaba además un modo conocido como ***Park Editor*** donde los jugadores podían crear sus niveles incorporando todo tipo de elementos de parques de patinaje.

Ilustración 12 - Tony Haw's Pro Skater 2 – Park Editor

En la línea de los RPG, aparece en junio de **2002 NeverWinter Nights** [22], publicado por Bioware [23] ya exitosa tras el título Baldur's Gate [24] (publicado en 1998). **NeverWinter Nights**, basado en las reglas de la tercera edición del juego de rol Dungeons & Dragons [25], incorporaba una herramienta de edición llamada Aurora Toolset [26] que permitía la creación de mapas o módulos completos del juego, permitiendo disfrutar como Dungeon Master. Esta herramienta era tan potente que CD-Projekt la utilizó para la creación de su primer videojuego **The Witcher** [27].

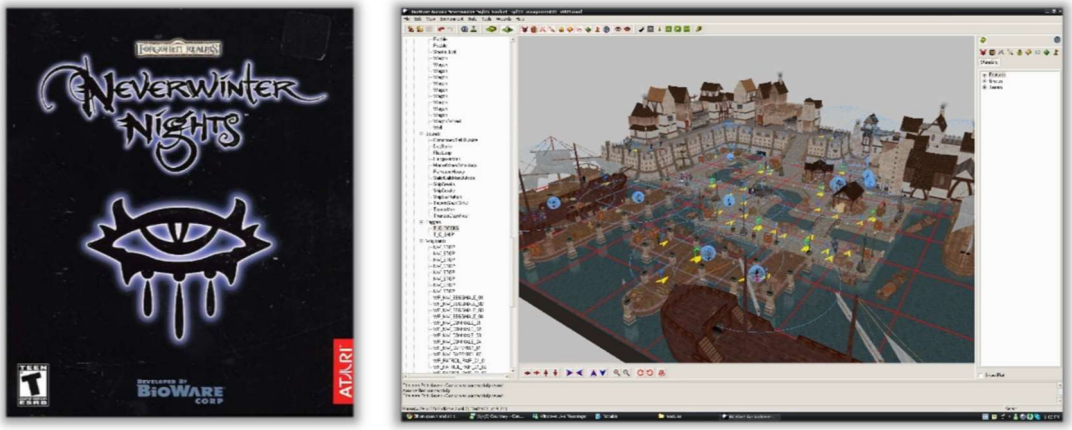


Ilustración 13 - Portada NeverWinter Nights y Aurora Toolset

En el mismo año, se publica **Wacraft 3: Reign of Chaos** [28]. Se trata de un juego RTS que continúa la exitosa saga de Blizzard Entertainment [29], con cambios importantes como el paso a gráficos 3D. **Warcraft 3: Reign of Chaos** incorpora un editor de niveles al igual que su predecesor, pero éste cobra especial importancia en esta edición, ya que permitía cambios incluso en la propia jugabilidad y estadísticas

de las unidades, dando lugar a nuevos modos de juego completamente diferentes al diseñado por la desarrolladora, hasta el punto de crear nuevos géneros en el sector como MOBA o Tower Defense.

Ilustración 14 - Portada Warcraft 3 Reign of Chaos y W3 World Editor

Ilustración 15 - League of Legends vs DotA 2

Defense of the Ancients o más conocido como **DotA** [30], es un mapa personalizado creado con el editor de niveles del sucesor **Warcraft 3: The Frozen Throne** [31], cuyo éxito dará lugar a la creación de juegos como **League of Legends** [32] (2009) o el sucesor como videojuego independiente en 2010 de la mano de Valve, **DotA 2** [33], ambos rivales en el género.

En **2004**, Valve lanza una herramienta de edición avanzada para su motor gráfico Source, **Garry's mod** [34] empleado para la creación de Half-Life 2 [35]. Permite la modificación de objetos y experimentación con las físicas de estos. Dos años

después, la herramienta era tan popular que se publicó como videojuego independiente.

Ilustración 16 - Garry's mod

En **2007** Bungie Studios [36] publica **Halo 3** [37]. El videojuego incorporaba un editor de niveles y modo de juego tan potente que hizo que el título fuese uno de los más exitosos y vendidos hasta el día de hoy, incorporando además la posibilidad de compartir tus niveles personalizados subiéndolos a una plataforma online desde la propia consola para permitir que cualquier persona en cualquier parte del mundo, pudiese disfrutar de tu creación al momento.

Ilustración 17 - Portada Halo 3

En 2009 se lanza públicamente un título, cuyo éxito será arrastrado hasta la actualidad, **Minecraft** [38]. **Es** un juego de construcción sandbox, sin ningún tipo de objetivo cuya jugabilidad se basa en la obtención de recursos por cualquier parte del mapa para así poder combinarlos y hacer realidad cualquier idea que se le pueda

ocurrir al jugador. El videojuego en sí mismo conforma una edición tan potente que jugadores han conseguido desarrollar dentro del juego funciones aplicables a la vida real tales como puertas lógicas [39] o incluso videollamadas [40].

Ilustración 18 - Minecraft

Similar a Minecraft, en 2011 se lanza otro videojuego con éxito, **Terraria** [41]. Se trata de un videojuego sandbox en 2D siguiendo la misma idea de **Minecraft**, pero siendo mucho menos potente que éste último, centrándose en otros aspectos como los jefes aleatorios, objetos potentes, equipaciones, armaduras, etc.

Ilustración 19 - Terraria

Ilustración 20 - Fortnite

Como éxito hasta la actualidad tenemos la publicación estrella de Epic Games [42], **Fortnite** [43], dentro del género de los Battle Royale, publicado en 2017 con múltiples similitudes a Player Unknown Battlegrounds [44] (PUBG), pero con dos factores diferenciadores, el estilo gráfico y la incorporación de un modo creación dentro del propio juego, convirtiéndolo en un título único dentro del género con más de 350 millones de jugadores.

En resumen, durante toda la historia de los videojuegos, se ha intentado eliminar la línea entre desarrolladores y jugadores, permitiendo a éstos últimos la posibilidad de dar rienda suelta a su ingenio y creatividad, facilitándoles las herramientas necesarias para poderlo llevar a cabo.

Como podemos comprobar, es una práctica que exitosa que ha ayudado a que nazcan títulos importantes de la talla de The Witcher, Garry's mod o incluso DotA entre muchos otros, que ha supuesto la creación de un género dentro de los videojuegos como es la creación de ciudades y que ha motivado a los estudios a incorporar toda esta parte creativa dentro de sus videojuegos como parte de la propia jugabilidad para ofrecer al jugador plena libertad.

2. Posición Home Factory

Home Factory está basado en la construcción en primera persona que incorpora diferentes géneros y técnicas con el objetivo de convertirse en un título diferenciador en la industria. En él, se combinan géneros como los sandbox, construcción y añadiendo un sistema de subastas para evaluar y mejorar la progresión del jugador permitiéndole realizar construcciones de mayor envergadura o disfrutar de sus creaciones.

En este trabajo se realiza un prototipo que incluye la combinación de estos elementos como primera iteración para el futuro desarrollo completo del videojuego. Uno de los principales objetivos de Home Factory es el de transmitir al jugador la sensación constante de tranquilidad y libertad y mejorar progresivamente el título tomando como referencia la propia comunicad de

jugadores, ampliando su contenido, mejorando e incluso visualizando diseños de otros jugadores que puedan servir como idea para futuras implementaciones.

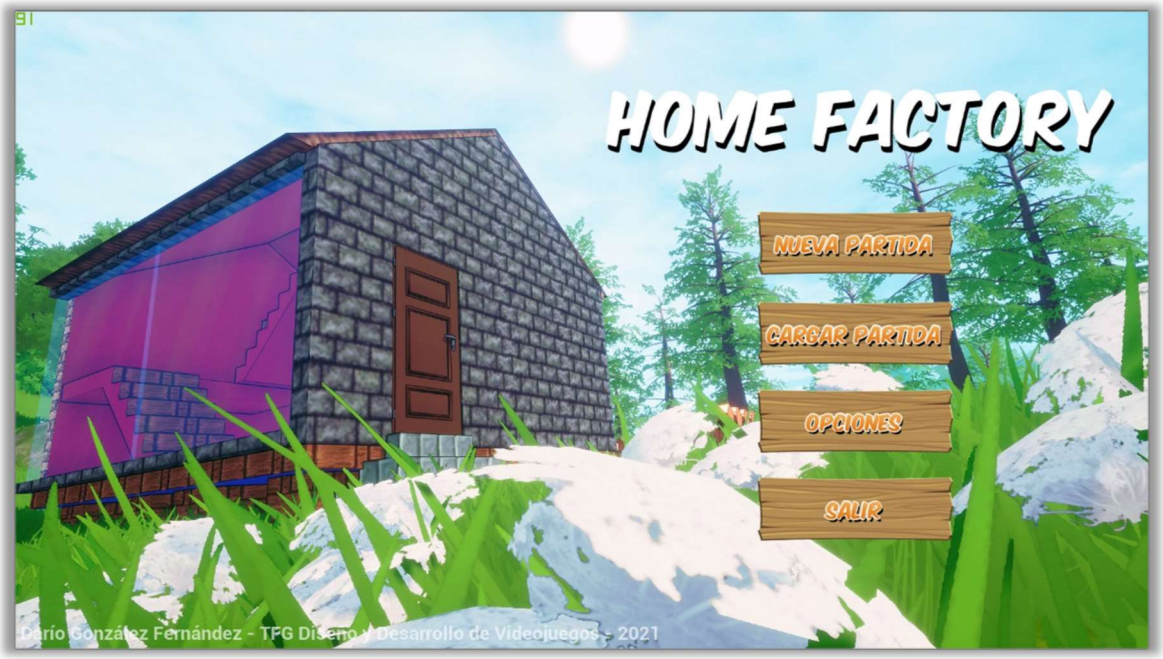


Ilustración 21 - Home Factory (Provisional)

3. Estado del arte

Existe una gran cantidad de alternativas a la hora de diseñar o crear niveles para videojuegos, e incluso videojuegos que incorporan funcionalidades de diseño de niveles. En este caso, nos centraremos en los títulos más recientes que incorporan en su jugabilidad la creación y/o edición de niveles y otros que, aunque no basen su jugabilidad en la creación, incorporan en la jugabilidad un sistema de subastas.

3.1. Videojuegos que incorporan creación de niveles

Prison Architect

Prison Architect [45] es un título de simulación de gestión y construcción de prisiones, de aspecto básico pero de profundas opciones para el jugador dentro del género de los sandbox. Como jugadores deberemos tener en cuenta todos los aspectos necesarios para el correcto funcionamiento de una prisión: celdas, cocinas, comedor, duchas, guardias, jardineros, doctores, psicólogos, etc. Además de hacer frente a los posibles planes de escape de los presos.

Ilustración 22 - Prison Architect y su editor

Rust

Rust [46] se trata de un videojuego que se lanzó inicialmente en modalidad de acceso anticipado en Steam y que ha ido evolucionando progresivamente según los comentarios de la comunidad, llegando a convertirse en uno de los referentes en el género de la supervivencia. La base principal del videojuego es la recolección de materiales que serán necesarios a lo largo del juego para crear nuevas herramientas, armas o armaduras para nuestro personaje para defendernos en un entorno multijugador, punto donde cobra especial importancia el apartado de creación de estructuras ya que, una vez nosotros abandonemos la partida, ésta seguirá su curso y deberemos tener un refugio donde dejar a nuestro personaje para que no perdamos nuestras pertenencias mientras estemos fuera de la partida.

Ilustración 23 - Creación de estructuras en Rust

Fallout 4

Fallout 4 [47] es un título de la saga Fallout que añade un nuevo sentido a la recolección de objetos a lo largo del mapa para utilizarlos como componentes de construcción y que en anteriores entregas quedaban destinados únicamente a ser tirados o vendidos por el jugador. En esta entrega podremos reunir diversos componentes para la creación de armas, servoarmaduras (como se conoce a las armaduras en el título), mesas de trabajo, asentamientos, torretas o incluso ciudades completas.

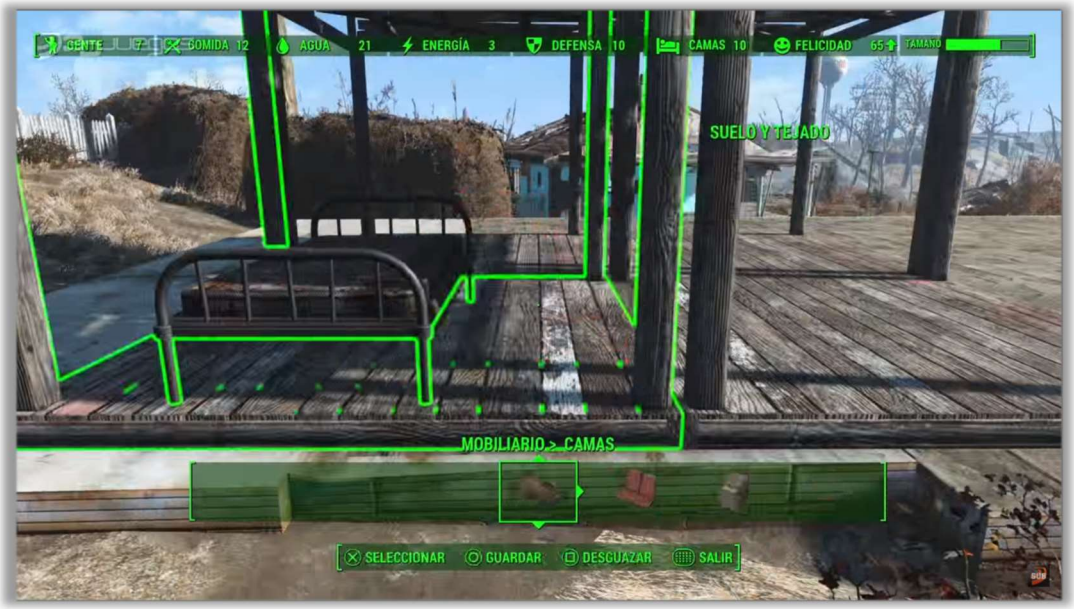


Ilustración 24 - Creación de estructuras en Fallout 4

No Man's Sky NEXT

No Man's Sky fue in título muy prometedor que terminó siendo muy criticado por los jugadores al no terminar cumpliendo las expectativas de la comunidad. Entre sus diferentes promesas, se encontraba la de poder encontrarse con otros jugadores en un entorno multijugador, algo que fue desmentido por los jugadores a los pocos días del lanzamiento y obligó a la compañía, Hello Games a proceder con más de cincuenta mil devoluciones. La compañía no abandonó al título, y éste ha seguido recibiendo actualizaciones desde su lanzamiento, acercándose cada vez más a lo que se había prometido a la comunidad.

Ilustración 25 - Creación de estructuras en No Man's Sky NEXT

Hello Games lanzó la actualización 1.5 llamada No Man's Sky NEXT [48] que incorporaba el sistema de creación de bases en cualquier punto del mapa, incluido bajo el agua, además de que todas las construcciones puedan ser visibles a otros jugadores en el juego.

ARK: Survival Evolved

ARK [49] es un juego lanzado en la plataforma de acceso anticipado de Steam donde apareceremos en la orilla del mar de una isla, con el único objetivo de sobrevivir a los peligros que tendremos a nuestro alrededor, desde hambre, sed, pasando por otros jugadores dispuestos a acabar con nosotros o incluso los dinosaurios que habitan la propia isla de Ark. Para protegernos de nuestro entorno, será necesario que creemos un refugio, que empezará siendo totalmente de paja hasta poder evolucionar a potentes construcciones, donde poder habitar.

Ilustración 26 - Creación de estructuras en ARK: Survival Evolved

Cities: Skylines

Cities: Skylines [50] es un simulador de creación donde tendremos que encargarnos de gestionar nuestra propia ciudad desde cero, empezando por elegir el terreno, teniendo en cuenta factores como recursos petroleros, minerales, agrícolas, forestales, conexiones con exterior, área edificable, zonas de agua, etc. Una vez seleccionado el terreno, tendremos que crear una red de carreteras para poder edificar junto a ellas, momento en que tendremos que empezar a cuidar de los recursos necesarios para los habitantes, tales como agua y luz.

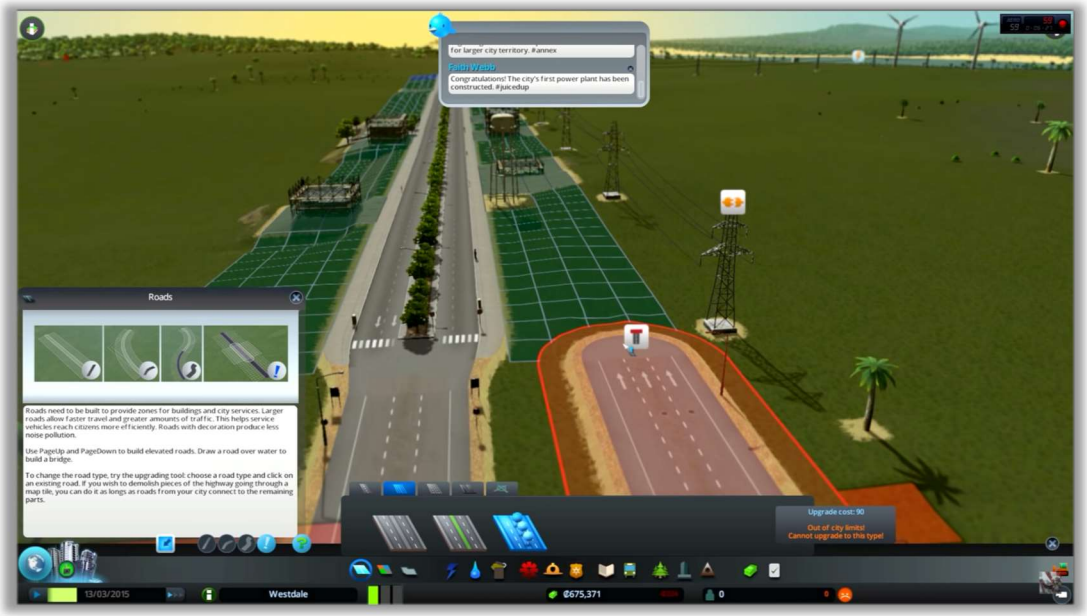


Ilustración 27 - Editor Cities: Skylines

El juego se puede considerar en sí un íntegro editor de niveles, donde nuestra jugabilidad está totalmente orientada a la constante construcción, edición y mejora de nuestra ciudad.

Stardew Valley

Creado por Eric "ConcernedApe" Barone, y basado en los principios de la saga Harvest Moon [51], Stardew Valley [52] intenta ser una versión completamente nueva y que subsana los errores que la saga había estado arrastrando entrega tras entrega. Stardew Valley nos pone en la piel de un personaje que está agobiado de su trabajo, de su ajetreada vida en la ciudad y decide mudarse a la granja que heredó de su abuelo para comenzar una nueva vida en Pueblo Pelícano. En la nueva granja se pueden desarrollar todo tipo de acciones, desde agricultura, ganadería, minería, pesca, recolección, etc.

Ilustración 28 - Construcción en Stardew Valley

El juego también constituye un editor completo de niveles ya que podremos ser capaces de editar todo nuestro alrededor para crear la granja que deseemos a nuestro gusto, mediante la creación de vallas, huertos, plantaciones, árboles, sistemas de riego automático, etc.

Animal Crossing: New Horizons

Animal Crossing: New Horizons [53] es un videojuego exclusivo de Nintendo y de su plataforma Nintendo Switch que forma parte de la saga aparecida por primera vez en la consola Game Cube, Animal Crossing.

Ilustración 29 - Selector de Terraforming en Animal Crossing: New Horizons

Se trata de un simulador en tiempo real que nos permite vivir en una pequeña isla donde tendremos que cuidar de la tierra, hacer recados a los vecinos, pagar hipotecas, mejoras en el pueblo, construcción de puentes, e incluso como novedad de esta entrega, la herramienta de Terraforming para la edición del propio terreno de forma horizontal y vertical y dar al jugador una capacidad de personalización de la isla al completo.

Ilustración 30 - Habitación personalizada en Animal Crossing: New Horizons

A diferencia de otros títulos que hemos visto anteriormente, la principal diferencia de Animal Crossing: New Horizons reside en la capacidad de personalización y decoración, tanto de la isla, como de nuestra propia casa y sus habitaciones, ya sea mediante alfombras, cuadros, mesas, lámparas, camas, tipos de suelo, paredes, etc.

Super Mario Maker 2

Super Mario Maker 2 [54] no podía faltar como un gran referente en el apartado de creación de videojuegos, y es que la secuela de Super Mario Maker, incorpora en esta ocasión además un modo historia, diseñado de forma muy inteligente ya que según vamos progresando dentro de la misma, la propia Nintendo nos irá dando ideas para construir nuestros propios niveles, desde una forma clásica característica de los juegos de Super Mario, pasando por scroll vertical, shoot 'em up, etc. Incorpora además un tutorial de 45 lecciones para intentar ayudarnos a pensar como un auténtico diseñador de niveles.

Ilustración 31 - Modo creación Super Mario Maker 2

3.2. Videojuegos que incorporan sistemas de subastas

Diablo III

Diablo III [55] incorporó uno de los sistemas de subasta más importantes hasta la fecha. Dentro del propio juego se incorporaban dos tipos de subasta, una hacía uso del oro, el dinero del juego, como moneda de cambio virtual. El segundo tipo, llamada Real Money Auction House [56] (RMAH) utilizaba el dinero real como moneda de cambio. Los jugadores podían hacer uso del dinero del que disponían

dentro de sus cuentas de Battle.net o incluso enlazar dichas transacciones con cuentas de PayPal.

Ilustración 32 - Casa de subastas de Diablo III

Finalmente, el sistema cerró sus puertas el 18 de Marzo de 2014, ya que los creadores consideraron que rompía la esencia del juego, el farmeo/looteo de objetos y equipación y trataron de compensar el cierre de la casa de subastas con el sistema llamado "Loot 2.0" que incorporaba más sentido e inteligencia a los atributos de los ítems que dejaban los enemigos al morir, en función de las características del personaje del jugador.

World of Warcraft

World of Wacraft [57], el título MMORPG más famoso de Blizzard, incorpora un sistema de subastas por oro del juego, donde es posible poner cualquier tipo de ítem o material a la venta, por la cantidad que el jugador desee (en función del mercado actual o los precios dentro del propio servidor). Resulta ser un sistema muy cómodo, ya que el jugador no tiene que estar constantemente pendiente de su objeto subastado. Una vez que lo pone a la venta, la próxima notificación que recibirá será cuando éste haya sido comprado por otro jugador, para poder recibir los beneficios de la subasta.

Ilustración 33 - Casa de subastas de World of Warcraft

Forza Horizon 4

En Forza Horizon 4 [58] se utilizan créditos (la propia moneda del juego), para comprar vehículos, mejoras de motor, frenos, neumáticos, y todo tipo de mejoras estéticas para la mejora de los coches. Incorpora además un sistema de compra/venta de vehículos, donde podemos indicar tanto el precio de salida, como el precio de compra directa.

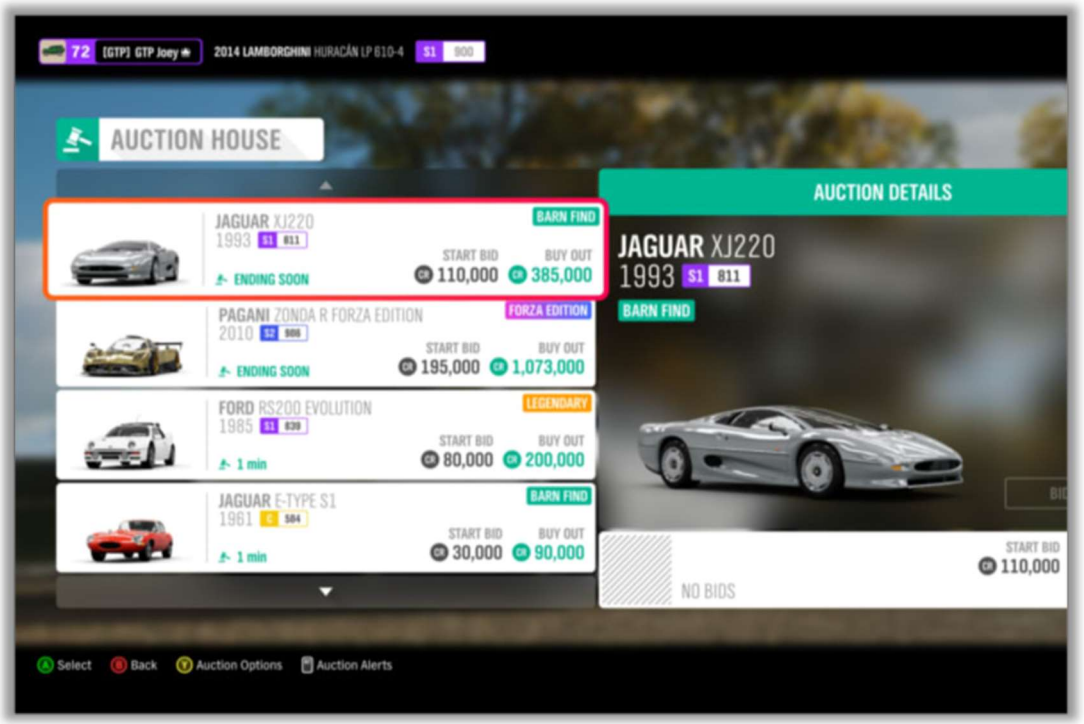


Ilustración 34 - Casa de subastas de vehículos en Forza Horizon 4

FIFA 20: Ultimate Team

Conocido comúnmente como FUT 20 [59], es el modo multijugador del famoso videojuego de Fútbol de EA Sports. Invita a los jugadores a crear un equipo profesional en base a la compra/venta de tarjetas y monedas virtuales.

El videojuego incorpora un sistema de subastas, donde podremos comerciar con las tarjetas que hayamos obtenido en sobres. En este caso también tendremos la opción de elegir la compra directa, o poder pujar por alguna de las opciones que estén disponibles en el mercado.

Ilustración 35 - Casa de subastas de FIFA 20: Ultimate Team

NBA 2K21

De la misma forma que sucede en FIFA 20: Ultimate Team, NBA 2K21 [60], título distribuido por 2K Sports, incorpora un sistema de subastas prácticamente similar al título de Electronic Arts.

Ilustración 36 - Casa de subastas NBA 2K21

3.3. Tabla comparativa estado actual

A continuación, podemos comprobar mediante una tabla comparativa los principales juegos del sector y las diferentes opciones que nos ofrece cada uno de ellos así como el posicionamiento de Home Factory dentro de la industria y su principal funcionalidad, la venta de estructuras mediante subasta.

Por otro lado, enfrentaremos aquellos títulos que incorporan dentro de su jugabilidad un sistema de subastas, frente al sistema ofrecido por Home Factory.

Título	2D/3D	Recolección	Jugabilidad basada en creación	Creación de estructuras	Edición de estructuras	Venta de estructuras
Prison Architect	2D	✗	✓	✓	✓	✗
Rust	3D	✓	✗	✓	✗	✗
Fallout 4	3D	✓	✗	✓	✗	✗
No Man's Sky NEXT	3D	✓	✗	✓	✗	✗
ARK: Survival Evolved	3D	✓	✗	✓	✗	✗
Cities: Skylines	3D	✓	✓	✓	✗	✗
Stardew Valley	2D	✓	✓	✓	✗	✗
Animal Crossing: New Horizons	3D	✓	✓	✓	✓	✗
Super Mario Maker 2	2D	✗	✓	✓	✓	✗
Home Factory	3D	✗	✓	✓	✓	✓

Tabla comparativa videojuegos basados en construcción

Título	Género	Transacciones online	Compra	Basado en mercado	Basado en programación
Diablo 3	ARPG	✓	✓	✓	✗
World of Warcraft	MMORPG	✓	✓	✓	✗
Forza Horizon 4	Conducción	✓	✓	✓	✗
Fifa 20: Ultimate Team	Deportes	✓	✓	✓	✗
NBA 2K21	Deportes	✓	✓	✓	✗
Home Factory	Sandbox	✗	✗	✗	✓

Tabla comparativa videojuegos que incluyen sistemas de subasta

3.4. Resumen comparativo Home Factory – actualidad

Podemos distinguir la comparación basada en dos puntos principales sobre las características del título.

Por un lado, nos encontramos con un elemento altamente común integrado en otros títulos, como puede ser la creación y edición de estructuras. **Sin embargo, no encontramos en ninguno de ellos la posibilidad de subasta de las mismas ni como elemento diferenciador del título, ni tan siquiera como una posibilidad**. En algunos de los títulos indicados anteriormente, cabe la posibilidad de deshacer construcciones que hayamos realizado, elemento que también forma parte de las mecánicas de Home Factory, sin embargo, no es considerado como una venta de elementos.

Respecto a la comparativa con otros juegos que incluyan sistemas de subastas, podemos comprobar según la tabla anterior que es único en el mercado en algunas características incorporando una especie de algoritmo donde, en función de ciertos parámetros de entrada (tales como la inversión que ha realizado el jugador en la casa junto con el valor de las piezas colocadas) se recibe la recompensa correspondiente, sin depender en este caso de la evaluación subjetiva de otros jugadores, centrándose por tanto de un enfoque a un modelo de un solo jugador.

Capítulo 2 - Objetivos

En este capítulo se tratará el problema a afrontar, cómo resolverlo, la aportación de ideas, la lista de objetivos y la planificación de trabajo.

Además de eso se incluye información sobre la metodología de trabajo empleada para obtener, a lo largo del desarrollo, todos los objetivos planteados.

2. Descripción del problema

Tras observar la situación del mercado actual se desea desarrollar un videojuego cuyo principal **elemento diferenciador** sea la **creación de estructuras** aplicando la **componente económica** sin que el jugador tenga que pensar en la recolección de materiales que, como se ha comprobado anteriormente, se trata de un recurso muy explotado en el mercado actual.

Se pretende además que el jugador tenga la libertad de **editar las estructuras** que va creando a su paso. Para ello se incorporarán interfaces gráficas de edición en cada uno de los componentes.

Por otro lado, y para potenciar esta componente económica se pretende que el jugador, una vez haya considerado acabada la construcción, sea capaz de recibir una recompensa por ello, subastando su estructura y consiguiendo por ello dinero del juego que le permitirá ampliar sus posibilidades tanto de creación como de edición, ya que cada uno de estos supondría un coste para el jugador.

2.1. Objetivos

El objetivo principal es:

- Desarrollar un prototipo de videojuego de construcción de estructuras, con mecánicas de subasta para la gestión de las mismas que se diferencie de videojuegos existentes según se ha expuesto en el apartado 3.

Los objetivos secundarios son:

- Desarrollar la funcionalidad de construcción y edición de estructuras

- Proporcionar al jugador la capacidad de gestionar su dinero inicial para crear y editar las estructuras
- Incorporar una interfaz de gestión atractiva e intuitiva para el jugador
- Asignar a la estructura del jugador una puntuación o "scoring", que determine valores de la estructura en la fase de subasta.

2.2. Estudio de alternativas

Tras haber establecido claramente los objetivos principales, cuyo principal componente era la diferenciación respecto al resto de videojuegos del sector que incorporasen técnicas similares, había que escoger una opción donde poder plasmar todas estas ideas.

Se estudió la posibilidad de diseñar el proyecto sobre Unity [61], sin embargo éste ofrecía un nivel de complejidad ligeramente superior respecto a su principal contrincante en el mercado, Unreal Engine [62], ya que éste está íntegramente enfocado al desarrollo en código C# mientras que Unreal Engine hace uso de C++, y lo más importante y elemento decisor, incluso incorpora la posibilidad de programación basada en Blueprints, que hace que todo este proceso resulte mucho más sencillo utilizando simplemente conocimientos de programación.

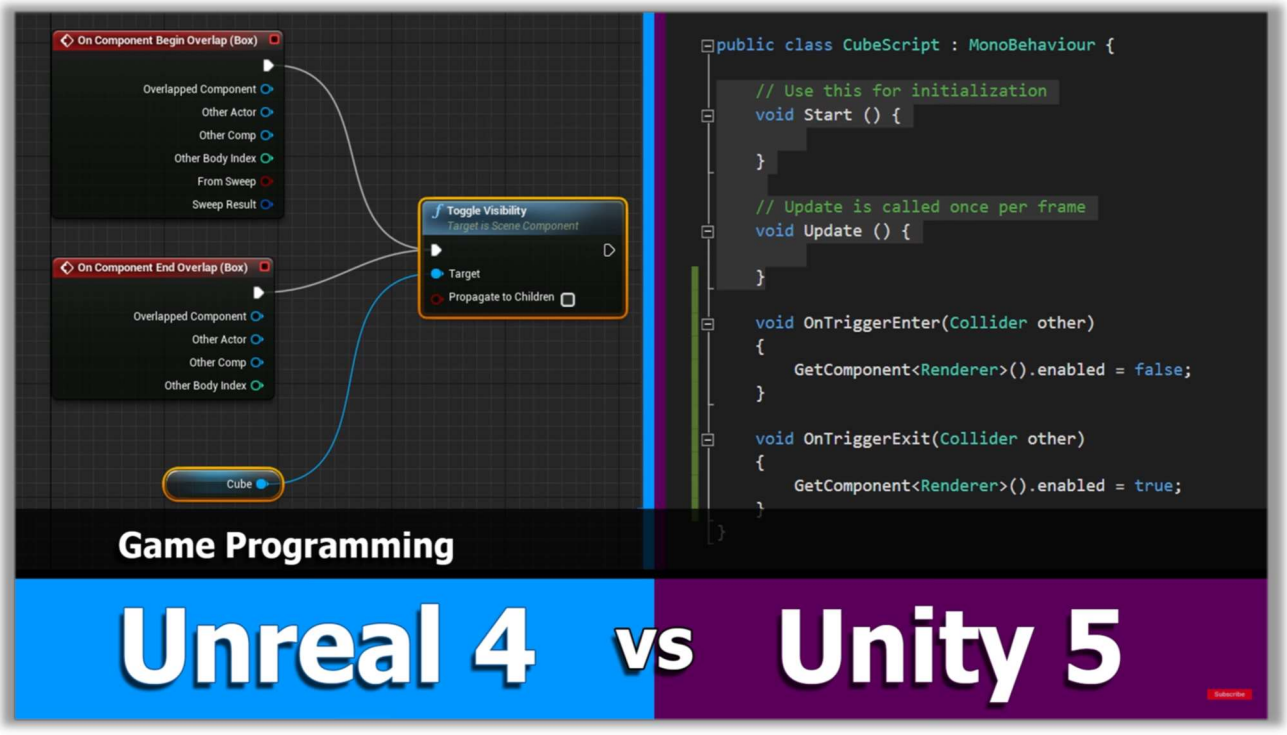


Ilustración 37 - Comparación Blueprints (UE4) vs. hardcoding (Unity)

2.3. Metodología empleada

La metodología necesaria para este proyecto era un tipo que permitiese el correcto flujo de trabajo, creación, desarrollo, evolución y mejora de pequeñas partes, haciendo el proyecto desde algo pequeño y continuar con su constante evolución. De esta forma, se podía comprobar progresivamente cómo los nuevos elementos interactuaban con los antiguos, y en caso de error o incompatibilidad, corregir a lo largo de esas iteraciones. Asimismo, se requería comprobar mediante la supervisión de la tutora que el proyecto sufría una evolución constante y que siempre incorporaba mejoras tras cada una de las iteraciones.

Por todas estas necesidades indicadas anteriormente se optó por una metodología de trabajo incremental, donde los cambios se realizan mediante un modelo basado en iteraciones. Cada incremento está basado en el conjunto de características a añadir progresivamente en el proyecto, comprobando que éstas funcionan correctamente y que no interfieren en el funcionamiento de las ya anteriormente implementadas.

Para esta metodología se han aplicado los siguientes objetivos en el proyecto de Home Factory:

- **Reunión con la tutora para decidir los objetivos base e iniciales del proyecto:** Reunión para comenzar la base principal de Home Factory.
- **Planificación establecida de dos semanas para completar la base del proyecto:** Se establece una fecha de dos semanas para la creación del prototipo base y funcional del proyecto.
- **Definición de los incrementos:** Una vez diseñada la base de proyecto, se decide junto con la tutora cómo van a ser los incrementos necesarios para que Home Factory evolucione progresivamente.
- **Desarrollo incremental:** Cuando los incrementos han sido definidos, se procede con el diseño de los mismos.
- **Validación:** Una vez finalizada la iteración, se comprueba que el incremento es perfectamente funcional.
- **Integración:** Una vez que se ha comprobado que el incremento es funcional, se integra con el resto de módulos ya existentes y se comprueba la

- **Repetición:** Una vez se ha integrado la última iteración y como los incrementos ya han sido previamente definidos se vuelve al punto de desarrollo, donde se continua con el incremento más adecuado a desarrollar y se repiten las fases **Desarrollo del Incremento**, **Validación de los incrementos** e **Integración**
- **Entrega:** Una vez el proyecto final ha sido implementado con todas las integraciones y se valida que todas las funcionalidades funcionan perfectamente entre sí, se procede con la entrega.

2.4. Planificación

La planificación se ha llevado a cabo siempre de forma conjunta y con la supervisión de la tutora, intentando plasmar las mejores ideas dentro del proyecto, a través de cuatro fases, de las cuales las tres últimas han sido ejecutadas en reiteradas ocasiones para llegar a implementar las funcionalidades necesarias y obtener la calidad deseada.

- **Fase de investigación:** Durante esta fase, y una vez decidida la aproximación que se quería llevar a cabo con un juego de construcción y subastas, se procede con la búsqueda de información, tanto histórica, como el estado del arte actual para contrastar nuestras ideas con el resto de los videojuegos disponibles en el mercado y poder así añadir pequeños factores que consigan hacer de Home Factory, un producto diferenciador.
- **Fase de diseño e implementación mediante iteraciones:** En esta fase, se definió el producto base y todas las peculiaridades que deberían ser implementadas en el título. Decisiones como el tipo de cámara, texturizado, estilo de juego, funcionalidades básicas, etc.

  Esta fase será ejecutada junto a las próximas dos en numerosas ocasiones, ya que una vez definida la base del videojuego, se realizaban ampliaciones del mismo mediante iteraciones basadas en pequeños cambios y funcionalidades y cómo debían encajarse éstas con las ya existentes.

- **Fase de pruebas y corrección:** Durante esta fase se testeaba Home Factory al completo. No era suficiente con que la propia base fuese completamente funcional, sino que se comprobaba de manera reiterada que las nuevas

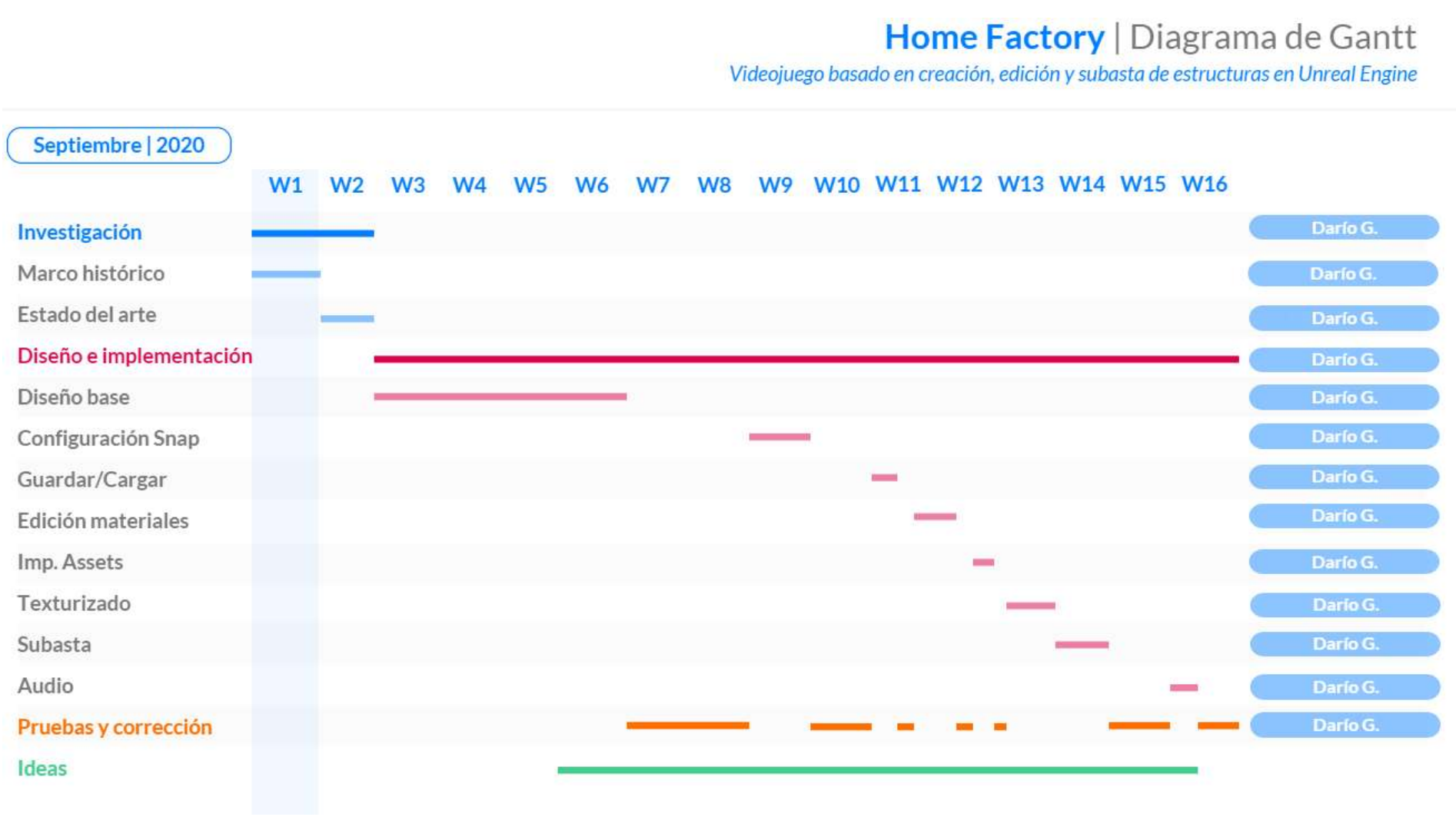
funcionalidades incorporadas no generasen ningún tipo de incompatibilidad con las ya existentes.

En caso de detectar algún tipo de error o incompatibilidad, la fase de pruebas y corrección se demoraba tanto tiempo como fuese necesario hasta proceder con la solución y la compatibilidad total de las nuevas implementaciones.

- **Fase de ideas:** En esta fase se generaban nuevas ideas para incorporar al proyecto. Funcionalidades como edición de materiales y cálculo de alturas, reembolso de piezas colocadas, interfaces gráficas, interacción con algunos elementos de las estructuras, gestión de transparencias, tipos de iluminación, etc.

Una vez que las ideas habían sido validadas, se volvía nuevamente a la **fase de diseño e implementación en sucesivas iteraciones**.

A continuación, se incluye un diagrama de Gantt explicando con mayor detalle el desarrollo del proyecto y cómo las fases anteriormente mencionadas han sido aplicadas en esta progresión.


Home Factory | Diagrama de Gantt
Videojuego basado en creación, edición y subasta de estructuras en Unreal Engine
Septiembre | 2020
W1 W2 W3 W4 W5 W6 W7 W8 W9 W10 W11 W12 W13 W14 W15 W16
Investigación
Marco histórico
Estado del arte
Diseño e implementación
Diseño base
Configuración Snap
Guardar/Cargar
Edición materiales
Imp. Assets
Texturizado
Subasta
Audio
Pruebas y corrección
Ideas
Darío G.

Capítulo 3 – Herramientas

En este apartado se hablará en detalle sobre las diferentes herramientas y tecnologías empleadas durante el desarrollo de Home Factory

3.1. Unreal Engine 4.24.3

Unreal Engine 4 (UE4) es un conjunto completo de herramientas de creación para desarrollo de videojuegos, visualización automovilística, arquitectónica, creación de contenidos para cine y televisión, producción de emisiones y eventos en directo, simulación y otras aplicaciones en tiempo real. Permite el desarrollo de proyectos en múltiples plataformas tales como Windows, PlayStation, Xbox Series X, Xbox One, Nintendo Switch, Google Stadia, MacOS, iOS, Android, AR, VR, Linux, SteamOS, HTML5, etc.

Unreal Engine ofrece dos tipos de licenciamiento dependiendo del uso final que se vaya a dar al contenido creado con su motor gráfico:

- **Licencia de Publicación:** La licencia es gratuita y supondrá un 5% de derechos de autor cuando el juego sea publicado, monetizado y genere unos ingresos brutos durante la vida del videojuego, que supere $1.000.000
- **Licencia Creadores:** La licencia es gratuita y 100% libre de derechos de autor. Podremos usar este tipo de licencia para crear proyectos internos, gratuitos, desarrollar contenido lineal o proyectos personalizados para clientes, siempre y cuando éstos no sean publicados.
- Se ofrecen además opciones de licenciamiento personalizado.

Su código fuente está escrito en C++, ofreciendo un gran rendimiento y además ofrece la posibilidad de programación mediante blueprints, que es la forma estándar para programar que incorpora Unreal Engine dentro de su entorno de desarrollo. Consiste en múltiples cajas interconectadas unas con otras que permiten todas las funciones posibles y necesarias dentro del mundo de la creación de videojuegos. No supone un sustituyente a la programación en C++. Estas cajas incorporan su propio código C++ en ellas de una forma predefinida (y editable) ofreciendo un entorno más sencillo, intuitivo y visual para desarrolladores que conozcan en profundidad cómo

funciona la programación, haciendo que ambos sistemas funciones perfectamente coordinados e integrados entre sí.

Se incluyen distintos tipos de blueprints [63] que servirán para ejecutar distintos tipos de funcionalidades. Los más importantes son:

- **LevelBlueprint:** Actúa como un gestor de eventos a nivel global dentro del nivel que esté actualmente. Cada nivel cuenta con su propio LevelBlueprint aunque no todos los niveles requieren de uno. Otra propiedad importante de éstos es que permiten animar propiedades de los actores con el tiempo o crear secuencias dinámicas en el juego.

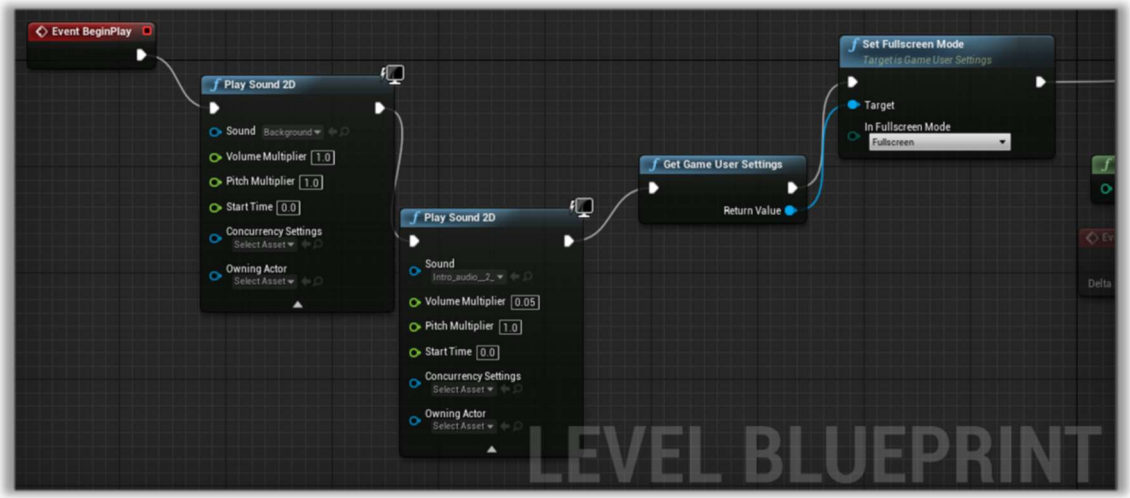


Ilustración 38 - Level Blueprint

- **Blueprint Class:** Permite a los desarrolladores agregar contenido y funcionalidades encima de las clases de juego existentes. Se especificará siempre la clase padre en la que se basará el blueprint, es decir, la clase de la que heredará todas las propiedades y de las cuales se agregarán funciones o serán sobreescritas y modificadas.

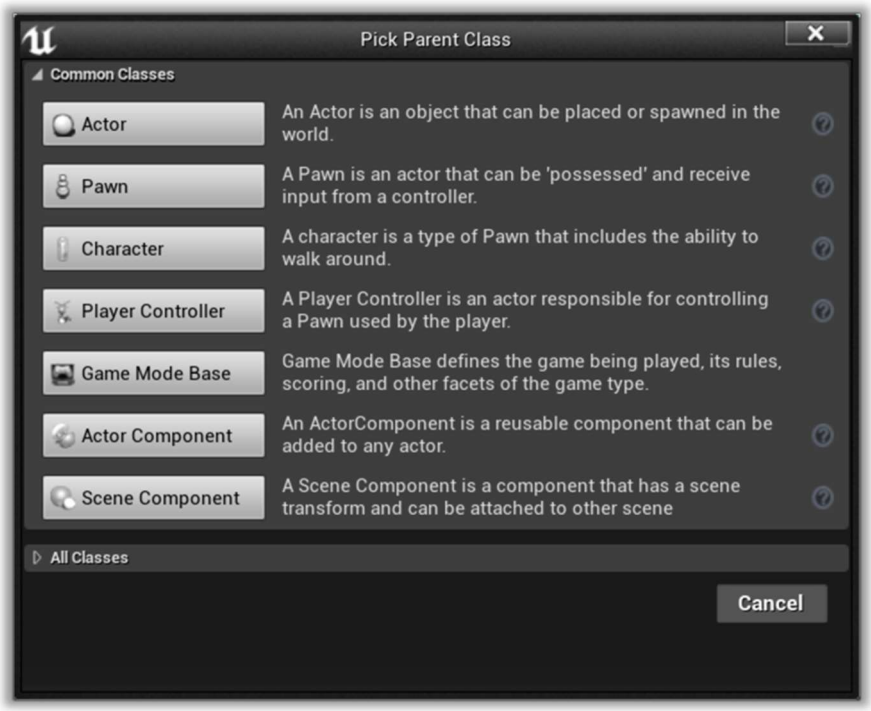


Ilustración 39 - Opciones de herencia Blueprint Class

Las clases padre predeterminadas más importantes son:

- Actor: Objeto que se puede colocar en el mundo. Se denomina actor pero no se refiere a un personaje, sino a cualquier elemento que pueda ser ubicado.
- Pawn: Es un tipo de actor determinado cuyo control puede ser absorbido y recibir la entrada de un controlador, ya sea un ratón, un mando, etc.
- Character: Peón que incluye la habilidad de caminar, correr, saltar, etc.
- GameController: Actor responsable de controlar un peón controlado por el jugador.
- GameMode: Este aspecto define el juego que se está jugando, reglas, puntuación, etc.

- **Blueprint Interface:** Es el conjunto de una o varias funciones que se puedan agregar a otros Blueprints. Permite el envío y la compartición de datos entre diferentes Blueprints.

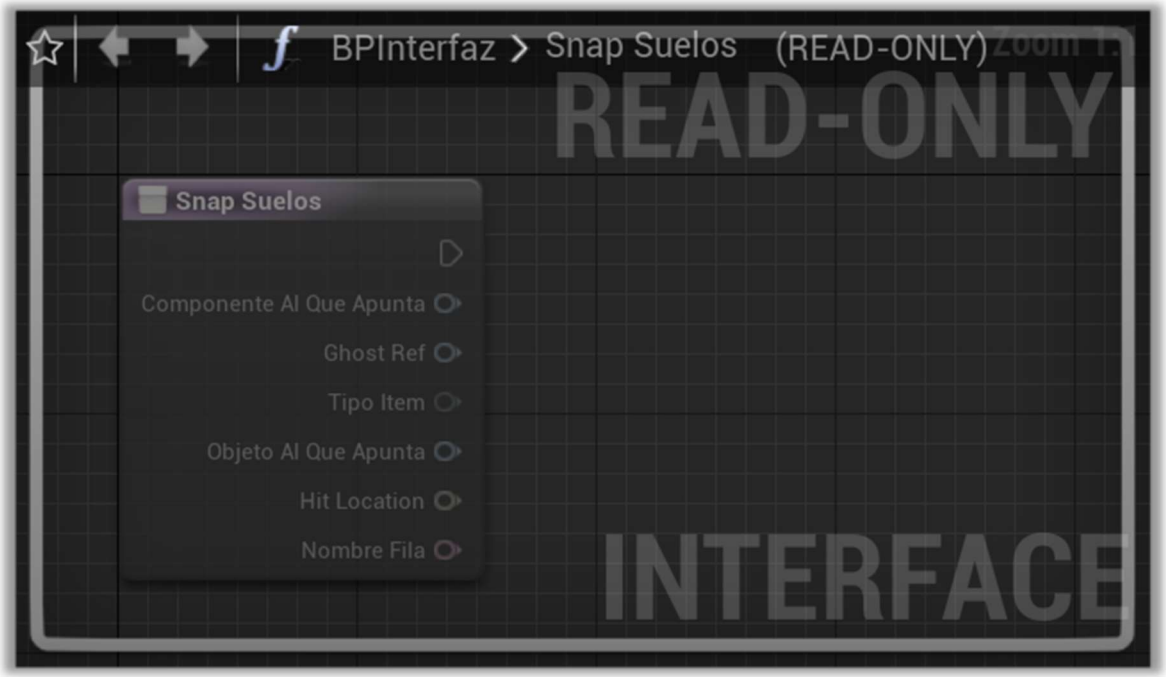


Ilustración 40 - Ejemplo de función en Blueprint Interface

- **Animation Blueprints:** Dedicados a controlar la animación de un modelo que posea una malla esquelética.

- **Widget Blueprints:** Con este blueprints podremos crear todo lo relacionado con UMG (Unreal Motion Graphics UI Designer [64]), desde menús interactivos para selectores de nivel, menús dentro del propio juego (menús de edición, inventarios, etc.), interfaces gráficas que nos indiquen el estado actual de lo que está sucediendo en el juego (dinero, salud, minimapa, munición activa, munición restante, arma activa, etc).

  En la opción que incorpora de Designer podremos ver una previsualización de cómo se verán dichas creaciones en la pantalla del juego en función de los parámetros de resolución que hayamos elegido.

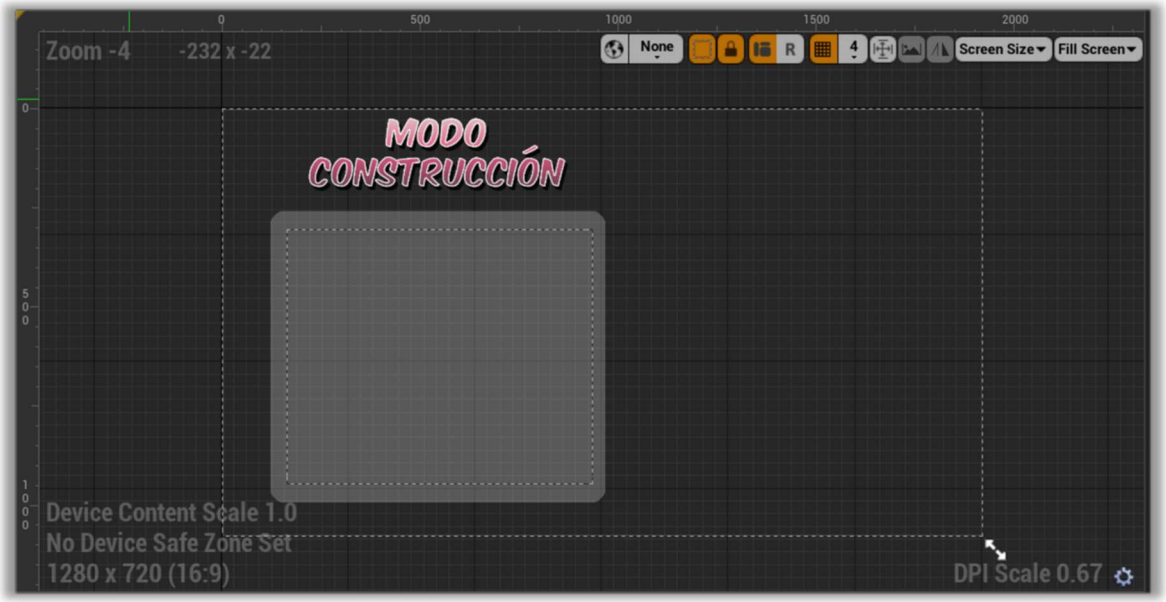


Ilustración 41 - Modo Designer en Widget Blueprint

Por otro lado, y en nuestro caso particular, el propio motor gráfico incorpora una serie de figuras geométricas básicas que pueden ser modificadas mediante las fórmulas de añadir y sustraer.

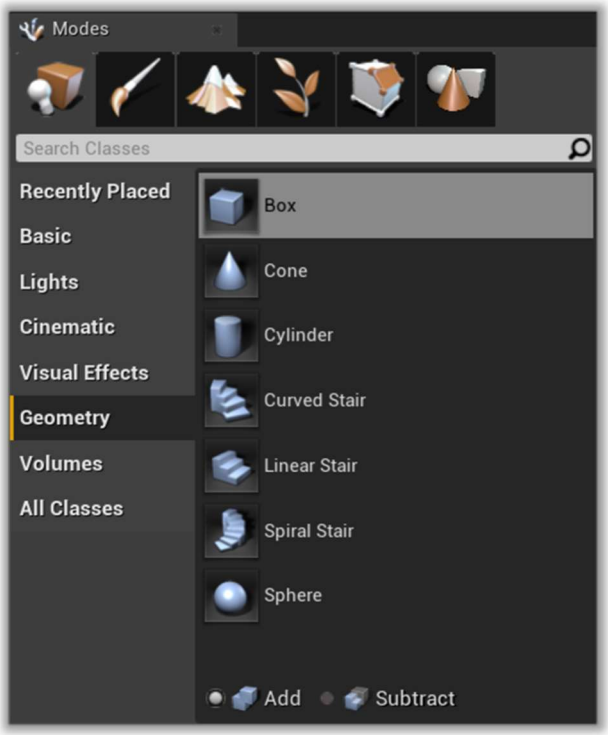


Ilustración 42 - Modo geometría UE4

Todos los elementos o componentes de estructuras han sido manualmente diseñados con esta utilidad.

Una vez que se colocan todos los elementos geométricos en el plano (tanto los que añaden componentes a la figura, como aquellos que utilizaremos para sustraer) y éstos son seleccionados al mismo tiempo, se puede generar una malla estática o static mesh. Una malla estática, es simplemente un modelo 3D que podemos agregar en cualquier parte de nuestro nivel. Sin embargo, para añadir eventos, programación y/o funciones, necesitaremos incorporar estas mallas estáticas dentro de un Blueprint.

3.2. Adobe Photoshop 2020

Photoshop [65] es un programa de pago de edición y diseño digital en 2D líder en la industria. Este software permite desde hacer montajes, editar, retocar, manipular y modificar cualquier tipo de imagen a través de las herramientas que ofrece el propio software, basado en un modelo de trabajo multicapa.

Unreal Engine, ofrece bastantes opciones a la hora de agregar efectos a textos y elementos de interfaz, sin embargo, se ha considerado Photoshop mucho más potente y adecuado para este tipo de trabajos.

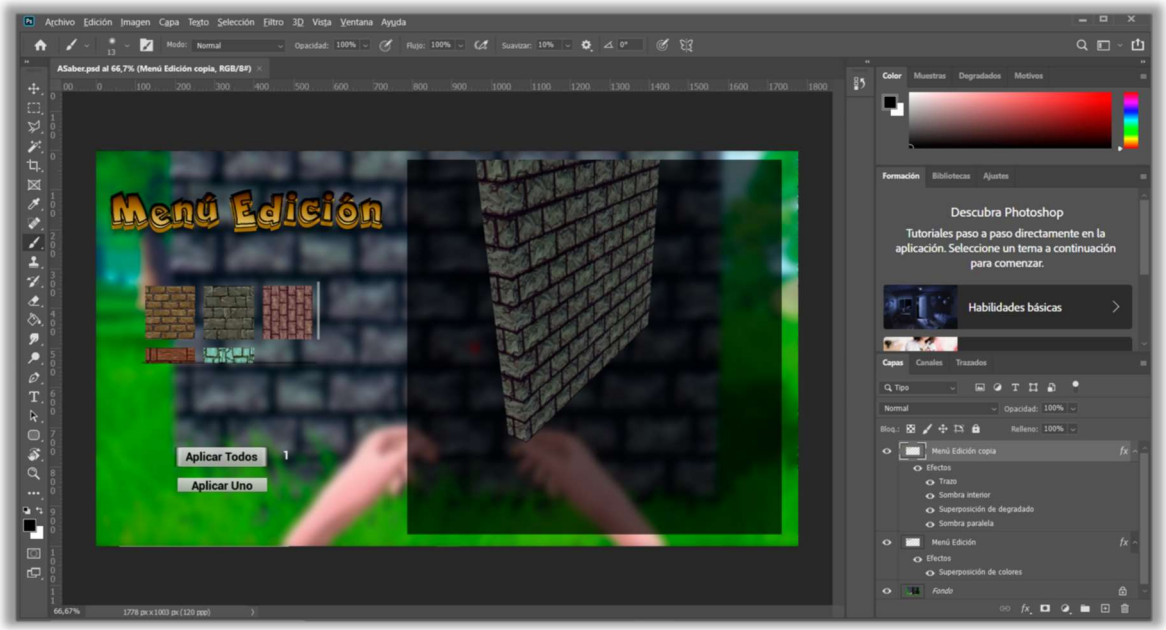


Ilustración 43 - Pantalla principal Photoshop

Photoshop ha supuesto una herramienta esencial en el proyecto, especialmente a la hora de editar texturas, crear botones, elementos que vayan a formar parte de la

interfaz, menús, etc., para hacer que todo ello resulte mucho más atractivo visualmente para el usuario final.

Además de lo ya mencionado, también se ha empleado como método para "diseño en vivo". Aunque Unreal Engine incorpora la herramienta designer dentro de widget blueprint, ésta no permite la visualización de los elementos en la propia pantalla de juego en vivo, por lo que muchas veces Photoshop ha supuesto el elemento diferenciador para poder probar distintos tipos de letras, botones, efectos, colores, sombreado, filtros, etc., que resultasen más atractivos para la interfaz de usuario de Home Factory.

3.3. UE4 Marketplace, UE4 AnswerHub y YouTube

Aunque no se trate de software como como tal, dichas herramientas han supuesto una gran fuente de conocimiento, ayuda y recursos especialmente importantes para el desarrollo de Home Factory.

UE4 Marketplace [66] es la tienda de assets para Unreal Engine. Aquí podremos encontrar assets de todo tipo, desde personajes, elementos para videojuegos en primera persona, efectos de sonido, de luz, animaciones e incluso escenarios y ciudades enteras para comprar.

Aunque los elementos básicos de construcción, programación, algoritmos, interfaces, etc., han sido diseñados manualmente, algunos elementos pertenecientes a Home Factory han sido directamente obtenidos desde UE4 Marketplace y modificados y adaptados al estilo visual y temática del videojuego. Se puede ver más información en el apartado **4.14. Fase 10 – Modificar aspecto visual de Home Factory**

UE4 AnswerHub [67] es sin duda una de las herramientas, junto con Youtube que más relevancia han tenido para el desarrollo del videojuego ya que se puede encontrar mucha información acerca de la propia funcionalidad del motor gráfico, funciones avanzadas, edición de funciones ya prediseñadas, etc.

YouTube [68] ha supuesto también una de las principales fuentes de información a nivel audiovisual, para la ampliación de conceptos básicos obtenidos durante la realización de la carrera e incluso la mejora a la hora de plasmar ideas de manera mucho más eficiente en el proyecto.

Capítulo 4 – Diseño e implementación

En este apartado se detallará el proceso completo de desarrollo de Home Factory basado en sus múltiples etapas, desde el punto base del proyecto, hasta el cierre del sistema de subastas final.

4.1. Definición de requisitos funcionales y no funcionales

Lo primero que se desea hacer, una vez con la idea clara de lo que se desea plasmar en Home Factory, es definir los requisitos funcionales y no funcionales. Para ello se han diseñado las siguientes tablas mostradas a continuación.

Requisitos funcionales

Nombre	**Descripción**
RF-1	La cantidad de dinero del jugador deberá ser mostrada en pantalla en todo momento para ser reconocida por el jugador.
RF-2	El manejo del videojuego se realizará mediante ratón y teclado, siguiendo el modelo de cualquier título shooter en primera persona.
RF-3	Se incluirá un menú donde se mostrará al jugador todas las piezas con las cuales puede crear estructuras.
RF-4	El jugador no podrá colocar una pieza cualquiera en cualquier zona aleatoria del mapa, sino que tendrá que colocar mínimo una pieza suelo donde sobre esta serán colocadas las demás.
RF-5	No todas las piezas pueden ser colocadas sobre otras del mapa, sino que cada una de ellas tendrá una disponibilidad concreta de ser colocada sobre otra pieza según se establece en la tabla "Interacción autoajustes" en la **ilustración 68**.
RF-6	Las piezas serán colocadas entre sí de forma automática mediante un sistema de autoajuste o "Snap-In". Todas las piezas deberán encajar entre sí perfectamente y al milímetro.
RF-7	Se mostrará al jugador una animación cada vez que coloque una pieza, simulando el pago de la misma y disminuyendo su contador de dinero.
RF-8	El jugador deberá disponer de un menú de edición donde será capaz de, mediante pago con la moneda del juego, cambiar el tipo de material que

	aplica a la pieza en cuestión, ofreciéndole la posibilidad de cambiar el material de una sola pieza, de todas las piezas del mismo tipo si el jugador está apuntando a la fachada de la estructura, o a todas las piezas que estén a la misma altura si el jugador está apuntando a una pared desde dentro de la estructura.
RF-9	Se habilitarán materiales bajo un candado, como desbloqueables o DLC para indicar al jugador la posibilidad de continuidad y contenido futuro de Home Factory
RF-10	Se habilitará una función de reembolso de piezas colocadas para que el jugador, en caso de haber colocado una pieza errónea, sea capaz de quitar esa pieza y por ello, recibir una recompensa económica parcial al gasto invertido
RF-11	La interfaz deberá mostrar sugerencias sobre cómo usarse y cuáles son los controles asociados a las funciones.
RF-12	El jugador deberá saber en todo momento dónde está apuntando, sin necesidad de incluir un puntero o crosshair en la interfaz.
RF-13	El videojuego deberá incorporar un sistema de guardar/cargar partida que permita al jugador guardar el estado actual y continuar en otro momento.
RF-14	El sistema deberá mostrar un mensaje en pantalla en caso de que el jugador se quede sin fondos e intente colocar piezas de estructura en el juego
RF-15	El menú de objetos, donde se muestren todos los objetos disponibles, deberá incluir el precio y una previsualización en 3D de la pieza sobre la que el jugador esté colocando el ratón.
RF-16	El menú de edición de objetos deberá mostrar una previsualización de la textura seleccionada en una figura 3D para que el jugador pueda ver el resultado antes de realizar la inversión

Requisitos no funcionales

Nombre	**Descripción**
RNF-1	La interfaz de usuario deberá ser simple, adaptable, informativa y estar en armonía con el diseño Cel Shading de Home Factory.
RNF-2	El juego deberá estar enfocado para todos los tipos de público. Por tanto debe ser sencilla de utilizar y atractiva para un público general.
RNF-3	El título deberá estar optimizado tanto a nivel de código como recursos gráficos para ofrecer el mejor rendimiento posible.
RNF-4	El videojuego debe de poder funcionar en la mayoría de las gamas de PCs posibles.
RNF-5	El juego debe de incorporar un sistema de reescalado y dimensionamiento adecuado para ajustarse correctamente al dispositivo donde se esté ejecutando.
RNF-6	El usuario debe de ser capaz de manejar lo básico del título en menos de 15 minutos

4.2. Definición de casos de uso

En cada diagrama de caso de uso se incorporará una descripción correspondiente a las funciones asociadas a cada interacción.

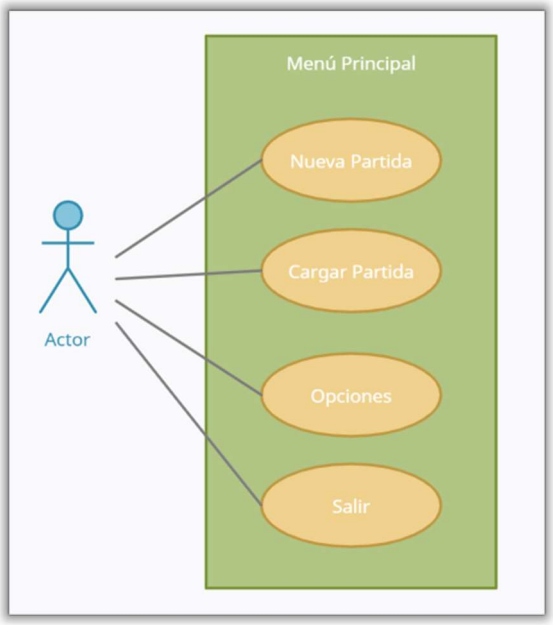


Ilustración 44 - Diagrama de caso de uso: Menú Principal

Menú Principal: El menú principal será la pantalla inicial que encontremos nada más cargar el juego. Desde este menú dispondremos de 4 opciones distintas:

- Nueva partida: Se cargará el mapa inicial donde el jugador podrá visualizar su personaje y podrá comenzar a jugar.
- Cargar partida: Se cargará la última partida que el jugador haya guardado.
- Opciones: Se mostrará la pantalla de opciones de resolución.
- Salir: Saldrá de la aplicación.

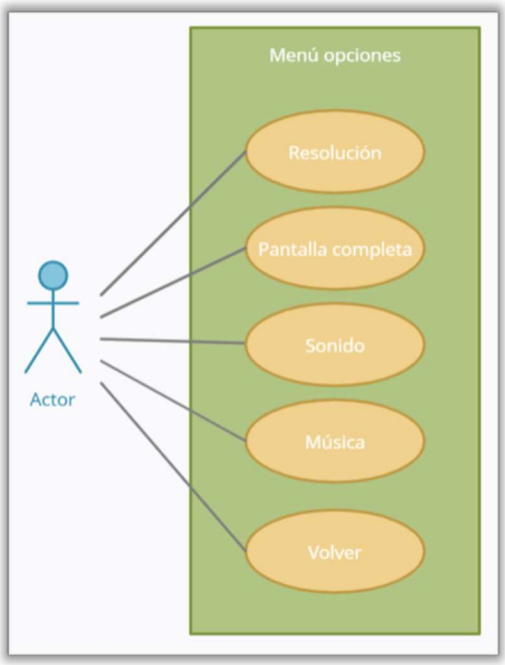


Ilustración 45 - Diagrama de caso de uso: Menú Opciones (menú principal)

Menú Opciones (desde menú principal)**:** El menú de opciones desde el menú principal, a diferencia del menú de opciones en partida, nos dará la posibilidad de modificar la resolución del juego e incluso activar y desactivar el modo ventana. Desde este menú dispondremos de 4 opciones distintas:

- Resolución: Ajustaremos la resolución de pantalla de nuestro juego para su óptima visualización.
- Pantalla completa: Podremos indicar si queremos el juego en pantalla completa o en modo ventana.
- Sonido: A través de una barra se podrá indicar el volumen de los sonidos del juego.

- Música: A través de una barra se podrá indicar el volumen de la música del juego.

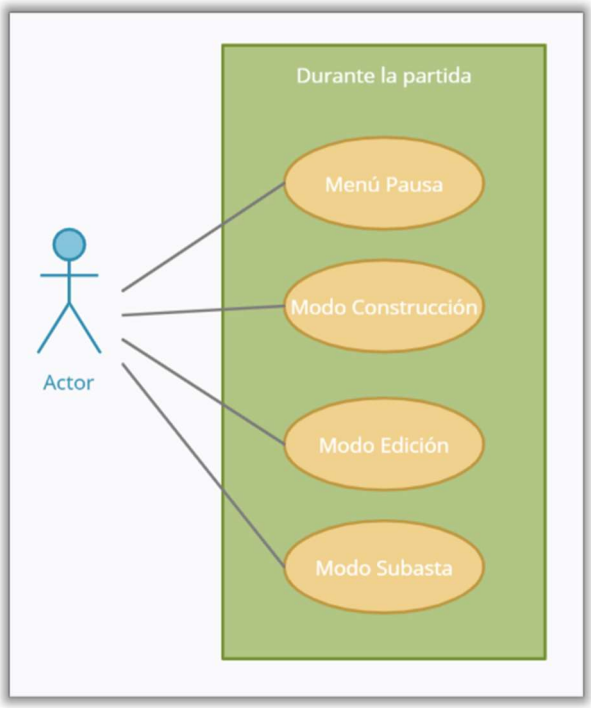


Ilustración 46 - Diagrama de caso de uso: Acciones disponibles para el jugador durante la partida

Durante la partida: En este diagrama se pretende describir las posibles funciones y menús interactivos con los que puede interactuar el jugador.

- Menú pausa: El jugador podrá acceder al menú de pausa de Home Factory.
- Modo construcción: El jugador podrá entrar en el modo donde puede seleccionar las piezas para posteriormente colocar en la escena.
- Modo edición: El jugador recibirá un menú donde se le mostrarán las texturas asociadas a la pieza a la que está apuntando para poder ser modificadas.
- Modo subasta: Una vez el jugador decida haber acabado con su construcción, podrá entrar en la pantalla de subasta de su creación.

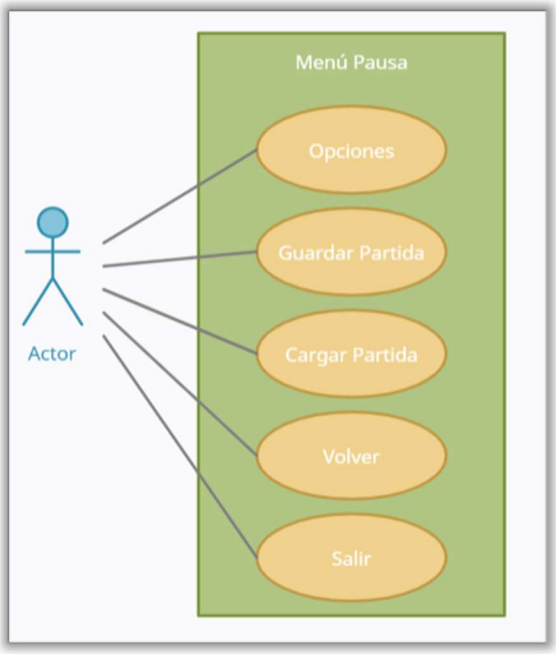


Ilustración 47 - Diagrama de caso de uso: Menú Pausa

Menú Pausa: Mientras que el juego se está desarrollando, el jugador podrá en cualquier momento entrar en el menú pausa y visualizar las siguientes opciones:

- Opciones: Se llevará al jugador a la pantalla de Opciones (durante la partida) diferente del menú opciones (desde menú principal)
- Guardar partida: El jugador podrá guardar el estado actual de la partida, las piezas, su colocación, el dinero, los materiales aplicados, etc.
- Cargar partida: El jugador podrá cargar el último estado guardado con el botón de guardar partida.
- Volver: El jugador volverá a la pantalla anterior, para poder continuar jugando.
- Salir: El podrá volver al menú principal (sin guardar estado actual).

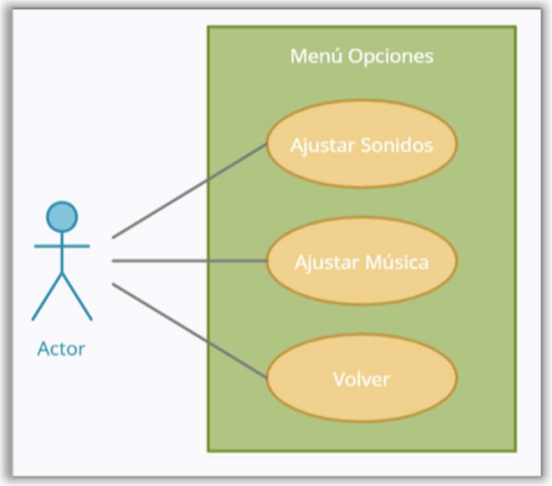


Ilustración 48 - Diagrama de caso de uso: Menú Opciones (durante la partida)

Menú Opciones: Se llevará al jugador a la pantalla de Opciones (durante la partida):

- Ajustar sonidos: A través de una barra se podrá indicar el volumen de los sonidos del juego.
- Ajustar música: A través de una barra se podrá indicar el volumen de la música del juego.
- Volver: El jugador podrá volver a la pantalla anterior (Menú Pausa)

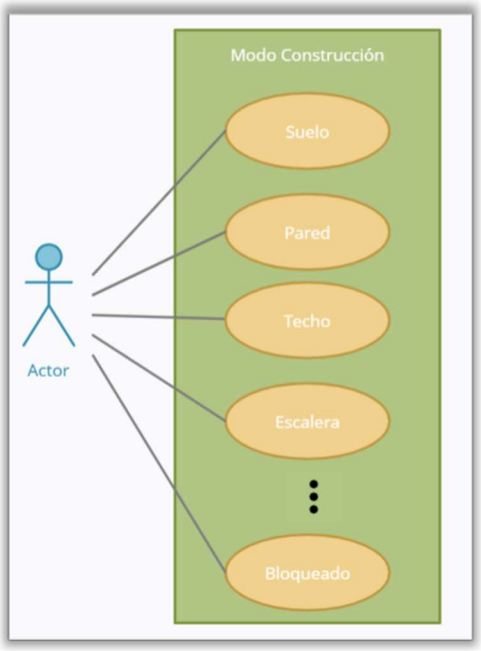


Ilustración 49 - Diagrama de caso de uso: Modo Construcción

Modo Construcción: Se mostrará al jugador todas las piezas disponibles para poder colocar en juego y poder comenzar o continuar con la construcción. El jugador entrará en este menú cada vez que quiera cambiar de tipo de pieza activa.

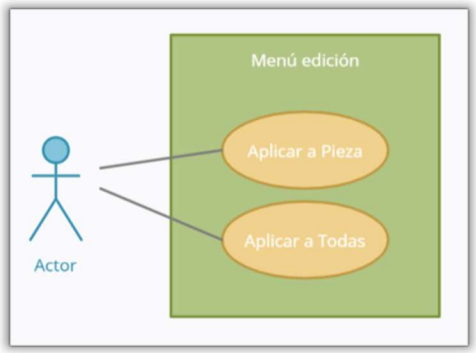


Ilustración 50 - Diagrama de caso de uso: Menú edición

Menú Edición: Cuando el jugador esté apuntando a una pieza ya colocada en el escenario, podrá lanzar el menú edición y se mostrarán en pantalla las distintas texturas que puedan ser aplicables a esa pieza. Si se trata de suelos, paredes (pared normal, pared puerta, pared ventana, etc.) se mostrarán dos opciones, en caso contrario, se mostrará sólo una opción:

- Aplicar a pieza: Se aplica la nueva textura a la pieza a la que el jugador está apuntando.
- Aplicar a todas: Si el jugador está apuntando a un suelo, se aplicará a todas las texturas suelo situadas a la misma altura. Lo mismo ocurrirá con las paredes siempre que el jugador esté apuntando a la parte interior de éstas.
  Si el jugador se encuentra apuntando a una pared desde el exterior, la función aplicar a todas, cambiará la textura de toda la fachada independientemente de la altura de la pieza a la que el jugador apuntaba.

4.3. Diagramas de flujo

Estos diagramas sirven para representar de forma gráfica los procesos de la aplicación desde el inicio hasta su finalización. Para ello se han utilizado las siguientes figuras para su representación:

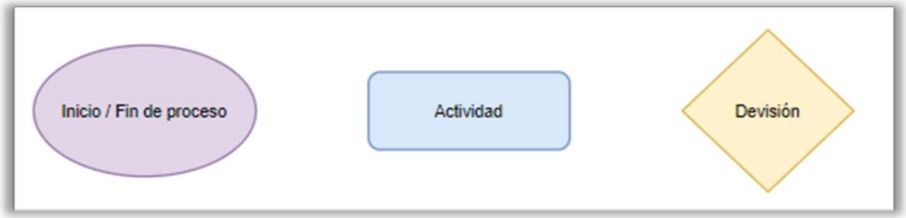
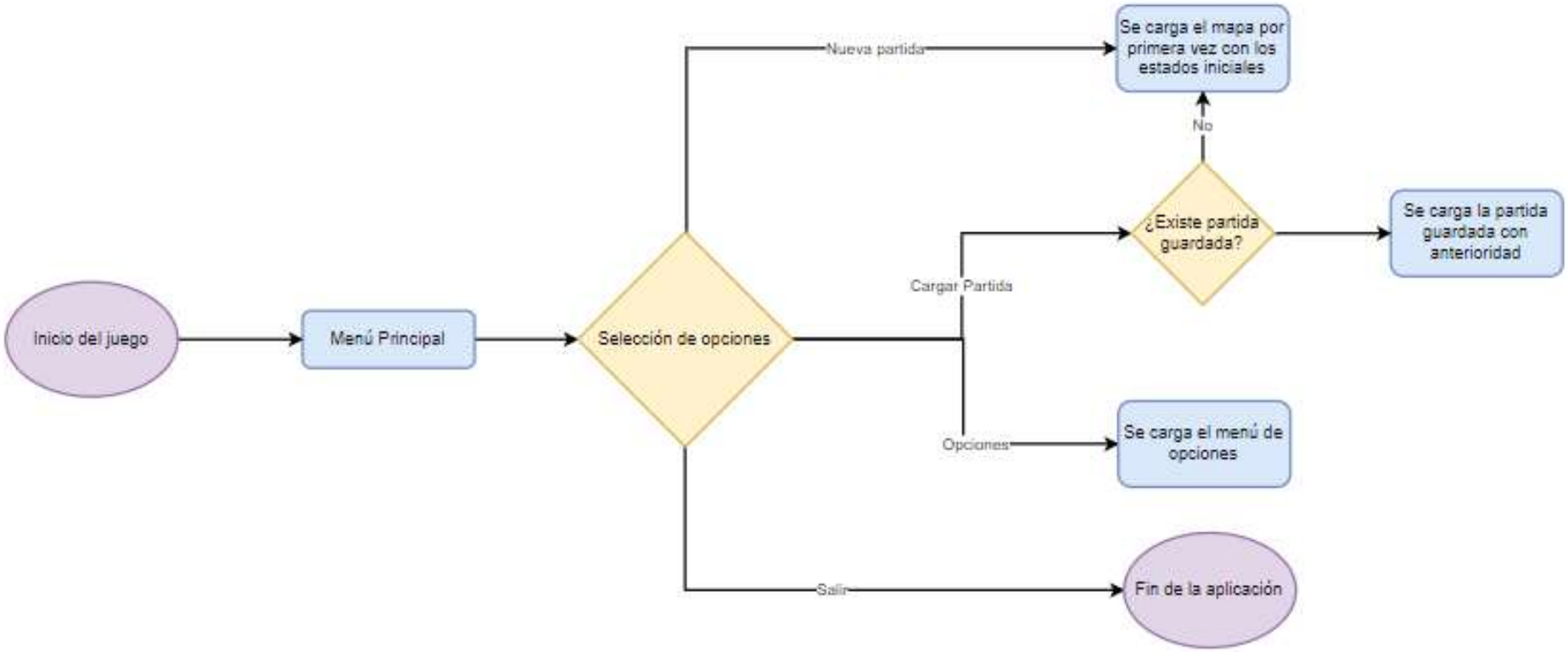


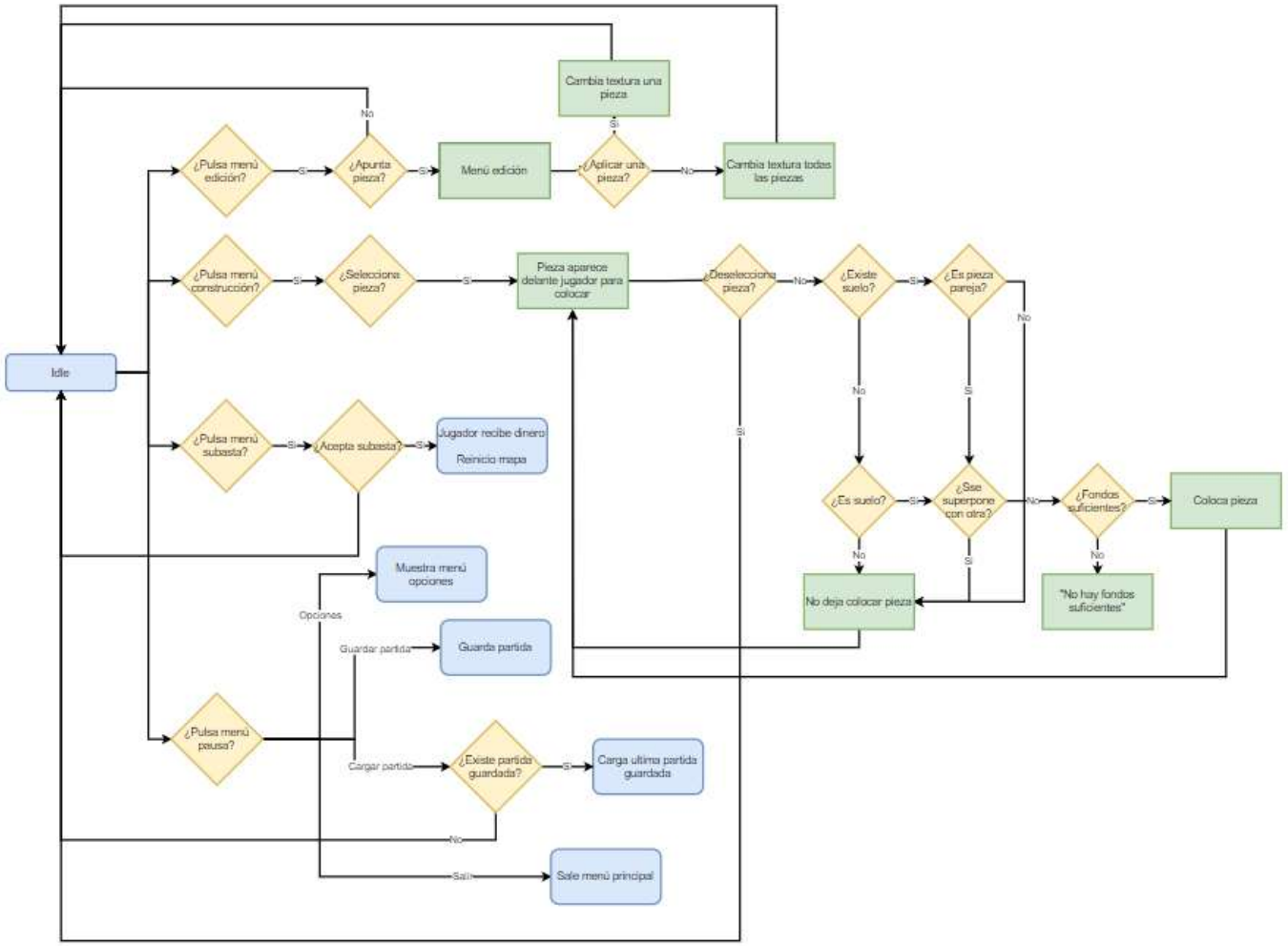
Ilustración 51 - Leyenda de diagramas de flujo

A continuación, se indican los distintos diagramas de flujo dependiendo del estado en el que se encuentre el jugador.

Menú Principal



Nueva partida / Idle



4.4. Diagrama de clases

A continuación, se procede con los diagramas de clases para dar la información más detallada posible sobre las clases principales utilizadas en Home Factory y las clases base de las que heredan en Unreal Engine.

Cabe destacar que debido al alto número de variables utilizadas, sólo se han incluido en este diagrama los métodos, distinguiendo entre métodos nuevos y los métodos override de clase padre (marcados con un símbolo @). Se recuerda que los métodos override son aquellos que ya están previamente definidos en la clase padre pero se desea añadir, restar o cambiar por completo la funcionalidad.

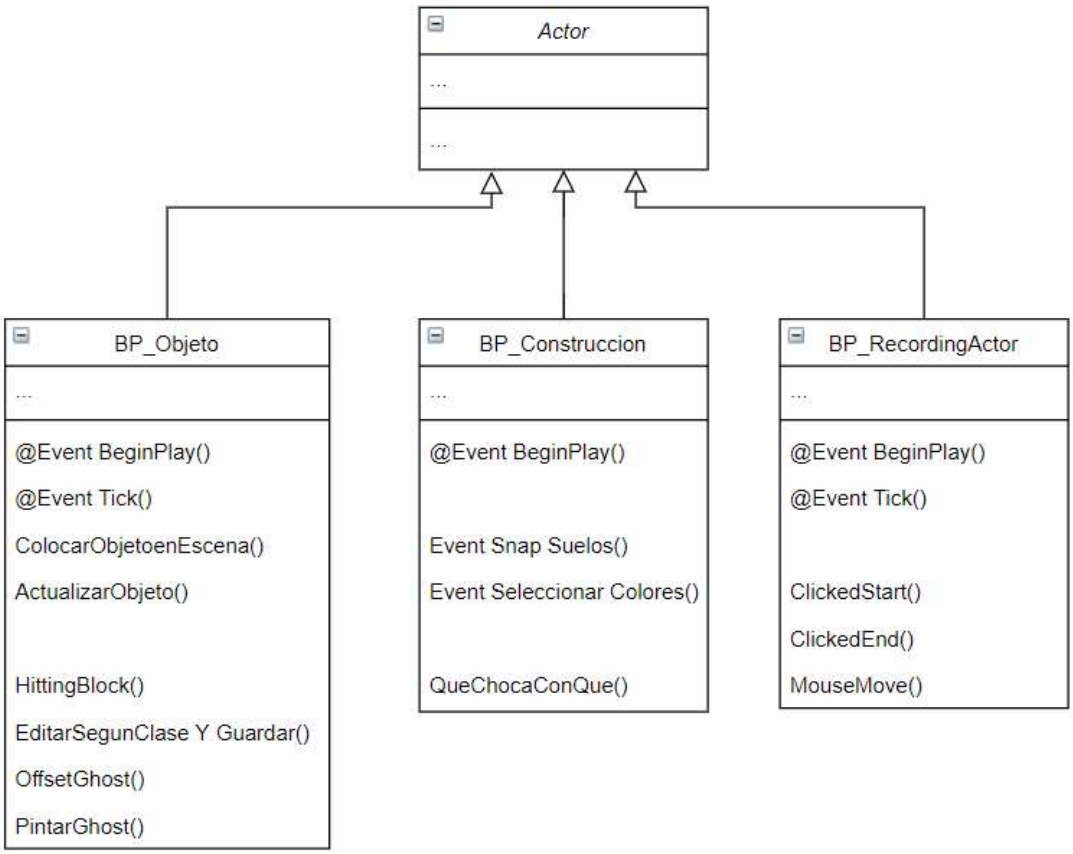
BP_Personaje

Personaje es una clase que hereda dentro de Unreal Engine de la clase Character.

- **Event BeginPlay():** Se ha realizado override en este evento para añadir la velocidad máxima de desplazamiento del jugador, recoger la instancia y crear los widget de estado, tanto el económico como la guía de interfaz.
- **Event Tick():** uno de los más importantes para la jugabilidad de Home Factory. Este evento es el encargado de realizar lo que esté escrito en el código por cada frecuencia de reloj, de forma ilimitada, por lo que se debe tener especial atención ya que se trata de un bucle infinito que se ejecuta constantemente mientras el juego (o el personaje en este caso) formen parte de la escena. En este caso, se ha limitado la comprobación constante de "Event Tick()" a la existencia de un Ghost en escena.

- **CrearGhost():** Es una función que será llamada inmediatamente después de la función de tecla de Tabulador (TAB). Una vez que el jugador abra el menú de creación de objetos y seleccione una de las piezas disponibles para construir. Primero se generará un objeto "vacío" inexistente y después, según la pieza que haya seleccionado el jugador, objeto realizará esta transformación en este objeto.
- **LeftMouseButton():** Realiza todas las comprobaciones pertinentes para verificar que el jugador pueda construir, es decir, que el jugador disponga de fondos, que la pieza seleccionada no esté haciendo overlap con otra pieza colocada en el mapa, etc.
- **RightMouseButton():** Comprueba que el elemento al que se está apuntando es una pieza de construcción y, en caso de serlo, la elimina del escenario y devuelve una cantidad de dinero proporcional al 65% del dinero invertido en la pieza (sólo en concepto de construcción) al jugador.
- **Key_Tab():** Tecla que despliega en pantalla el widget "SelectorObjetos" para poder seleccionar piezas para poder construir.
- **Key_ESC():** Tecla que despliega en pantalla el widget de "MenúPausa"
- **Key_E():** Tecla que despliega en pantalla el widget "BluePrint Colores" para poder cambiar la textura de la pieza seleccionada
- **Key_1():** Tecla que activa/desactiva el widget de ayuda para mostrar los controles al jugador.
- **Key_5():** Tecla que activa el menú de subasta
- **Key_Q():** Tecla que elimina la pieza que tengamos seleccionada. Elimina el Ghost que esté activo en ese momento.
- **Mouse_WheelUp/Down():** La funcionalidad implementada en la rueda del ratón hará que la pieza pueda rotar hacia la derecha o izquierda. En casos concretos como la escalera servirá para rotar (una vez apoyada) +/- 90º mientras que en la Teja transformará esta pieza en la TejaIder a TejaDizq.

BP_Objeto, BP_Construcción, BP_RecordingActor



Todas estas clases heredan de la clase Actor.

BP_Objeto

- **Event BeginPlay():** Se realiza un override para recoger la instancia y establecer la variable de fila seleccionada, es decir, cuál es el objeto que ha seleccionado el jugador y en el que "Objeto" tendrá que convertir su "malla estática" no visible.
- **Event Tick():** Override del evento dónde se comprueba constantemente el tipo de objeto en el que "objeto" se está convirtiendo para así hacer una llamada a la tabla de Offset de Corrección (Tabla que determina las posiciones en X, Y, Z adecuadas para que las piezas encajen perfectamente unas entre otras) que se aplican a cada una de las piezas.
- **ColocarObjetoenEscena():** Se almacena el valor de la pieza y se descuenta el valor de la pieza de los contadores de InstanciaJuego. Además se rellena el array de actores colocados.
- **ActualizarObjeto():** Según la pieza que haya sido seleccionada, se transforma el objeto ghost en esta figura.

- **HittingBlock():** Comprueba el estado de la pieza para verificar si se puede colocar o no.
- **EditarSegúnClaseYGuardar():** Asigna a las piezas un nombre para meterlas dentro del array de actores colocados y tener referencias de su posición, su textura, su valor, etc.
- **OffsetGhost():** Función que sirve para recibir la clase de la pieza seleccionada.
- **PintarGhost():** Asigna un color a la transparencia Ghost según el estado de la pieza. Si la pieza es verde, puede ser colocada en ese momento, si la pieza es amarilla, está en el aire, si la pieza es roja, está haciendo overlap con otra pieza o directamente no se puede colocar.

BP_Construcción

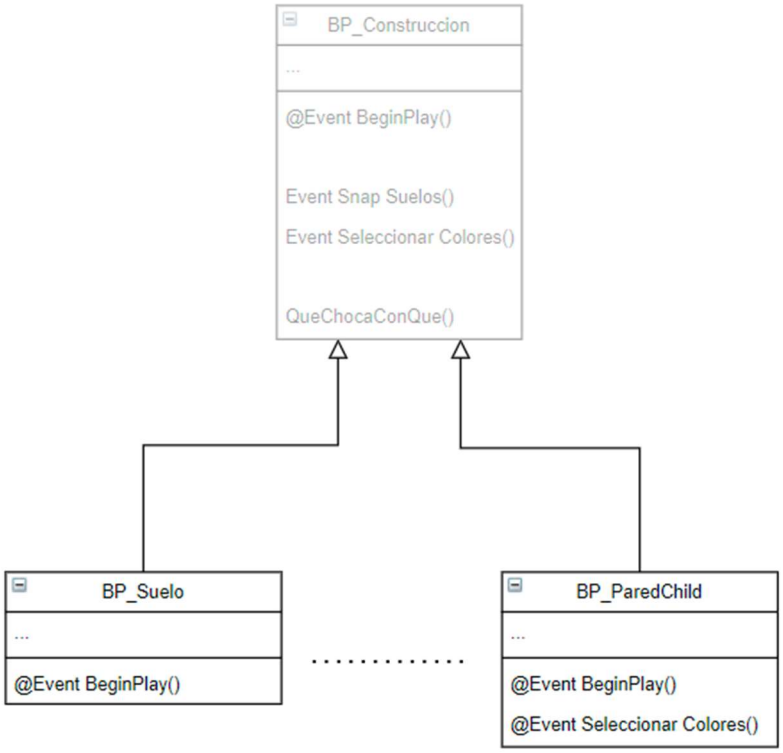
- **Event BeginPlay():** Recoge la instancia y contiene un timeline para la animación de escala de las piezas que se coloquen en el escenario.
- **Event Snap_Suelos():** Evento donde se recoge el ghost actual seleccionado, el componente al que apunta, el tipo de item, la posición, el nombre del objeto activo para después comprobar en la función **QueChocaConQue()**
- **Event SeleccionarColores():** Activa la interfaz correspondiente para seleccionar colores, enviando el índice del componente que va a ser modificado.
- **QueChocaConQue():** Evento que, como su propio nombre indica, recibe la pieza seleccionada con la pieza a la que está apuntando el jugador para así determinar cuál será su posición de Snap-in.

Se incluye una referencia a la clase de LibreríaAjustes que determinará cuál es el comportamiento de la pieza en función de la pieza seleccionada con la pieza de destino.

BP_RecordingActor

- **Event BeginPlay():** Recoge la instancia y el componente y/o componentes con tag “Record” para almacenarlos en un array donde serán posteriormente pintados en la previsualización 3D.
- **Event Tick():** Evento que aplica una rotación lenta y automática a la pieza seleccionada para que se pueda visualizar una rotación sobre el eje Z.

- **ClickedStart()/ClickedEnd():** Comprueba si la pieza de previsualización está siendo clicada para poder ser movida o no.
- **MouseMove():** En caso de que la pieza esté siendo clicada, indica la rotación que se aplicará a la pieza en el eje Z.



Tanto BP_Suelo, BP_ParedChild como casi todas las piezas de construcción heredan de BP_Construcción, excepto aquellas que son del tipo pared.

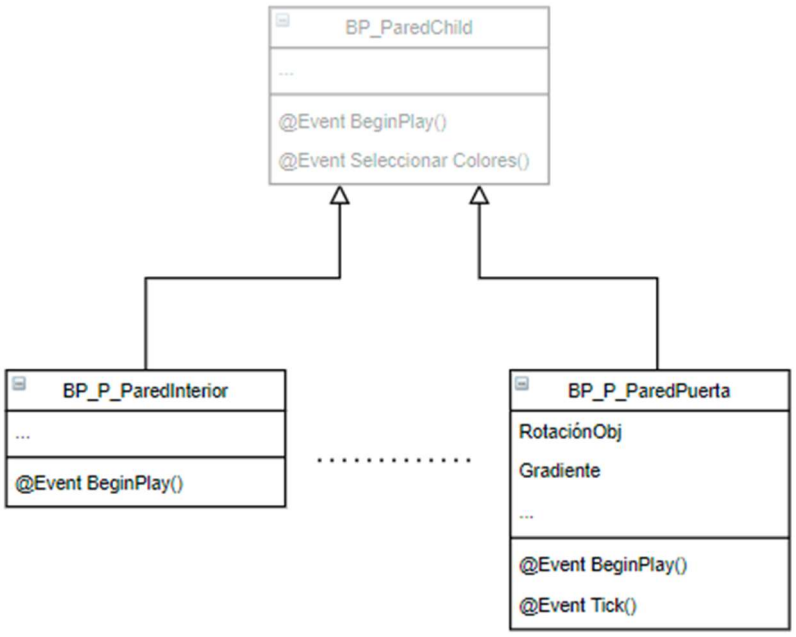
BP_Suelo

- **Event BeginPlay():** En este caso sólo haremos override de la función EventPlay() para que cada vez que se cree la pieza, se le asigne que es de tipo "Suelo" para futuras utilidades

BP_ParedChild

- **Event BeginPlay():** Al igual que en la clase Suelo, haremos override de la función EventPlay() para que cada vez que se cree la pieza, se le asigne se le asigne el nombre de tipo "Pared" para futuras utilidades.
- **Event SeleccionarColores():** A diferencia de las piezas de construcción, en este caso tenemos que tener en cuenta que partimos de dos índices para modificar el color. El índice 0 y el índice 1 dónde se corresponderá con la parte externa y

la parte interna de la pieza respectivamente para seleccionar dónde se aplicará la textura seleccionada.



Todas las piezas de tipo pared heredan de la clase BP_ParedChild.

BP_P_ParedInterior

- **Event BeginPlay():** Al igual que en la clase ParedChild, haremos override de la función EventPlay() para que cada vez que se cree la pieza, se le asigne el nombre de tipo "ParedInterior" para futuras utilidades

BP_P_ParedPuerta

- **Event BeginPlay():** Al igual que en la clase ParedChild, haremos override de la función EventPlay() para que cada vez que se cree la pieza, se le asigne el nombre de tipo "ParedPuerta" para futuras utilidades
- **Event Tick():** Al tratarse de una pared puerta, heredad de la clase pared y agrega la funcionalidad de apertura de puerta. De forma de que cuando nuestro personaje haga un overlap con la zona de bloqueo de la puerta, la puerta se abrirá de forma automática.

  Aplicamos la misma respuesta al resto de blueprints interior/exterior.

Una vez tratadas las clases más importantes del proyecto, se procede a hablar de la implementación de fases basada en iteraciones y los detalles realizados en cada una de ellas.

4.5. Implementación de funcionalidades

4.5.1. Fase 1 – Base de un proyecto de creación de estructuras

Para proceder con la implementación de un sistema de creación de estructuras, primero había que definir las piezas básicas que iban a ser utilizadas en la jugabilidad y proceder con su creación.

La mayoría de las piezas son geometría básica, por lo que podíamos proceder con la realización de todas ellas con la ya mencionada anteriormente herramienta que incorpora Unreal Engine de edición de Geometría.

A continuación, hablaremos a modo de ejemplo de la creación de las piezas más características incluidas en el videojuego:

- **Suelos**: Aunque haya otras piezas, como los techos, que tendrán una doble funcionalidad, ya que serán techo de planta y suelo de la siguiente planta a su vez, nos dirigiremos a suelo como la pieza clave que servirá de cimientos para el resto de la estructura ya que, sin esta pieza colocada en el juego, ninguna de las piezas disponibles podrá ser colocada.

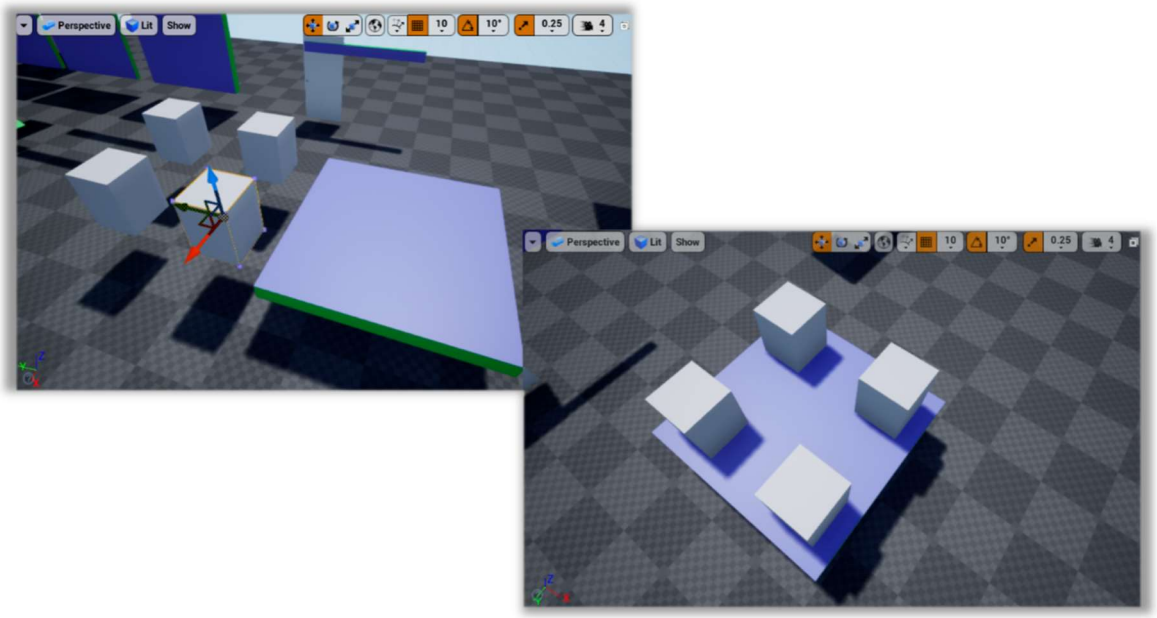


Ilustración 52 - Proceso de creación de suelo

La creación del suelo resulta bastante sencilla, ya que sólo consta de 4 bloques de idénticas proporciones y un cuadrado donde se apoyarán todas estas piezas.

Hay que tener especial atención con material que se asigna a cada una de las caras y piezas que compongan la figura antes de convertirlo en una malla estática, ya que cada material independiente supondrá una asignación posterior a un material, único o repetido. Si toda la pieza conserva el mismo material al convertirlo en una malla estática, sólo se podrá asignar un único material a toda la figura.

- **Pared puerta / Pared puerta interior:** Pieza de la estructura que, como su propio nombre indica, es una pared con un hueco de puerta. Sin embargo esta pieza esconde varias peculiaridades.
  La construcción de la pieza es bastante sencilla, se toma la misma pieza que se utilizó para la creación de la pared básica, y se agrega un rectángulo justo a la mitad que sirva de resta, para hacer el hueco de puerta.

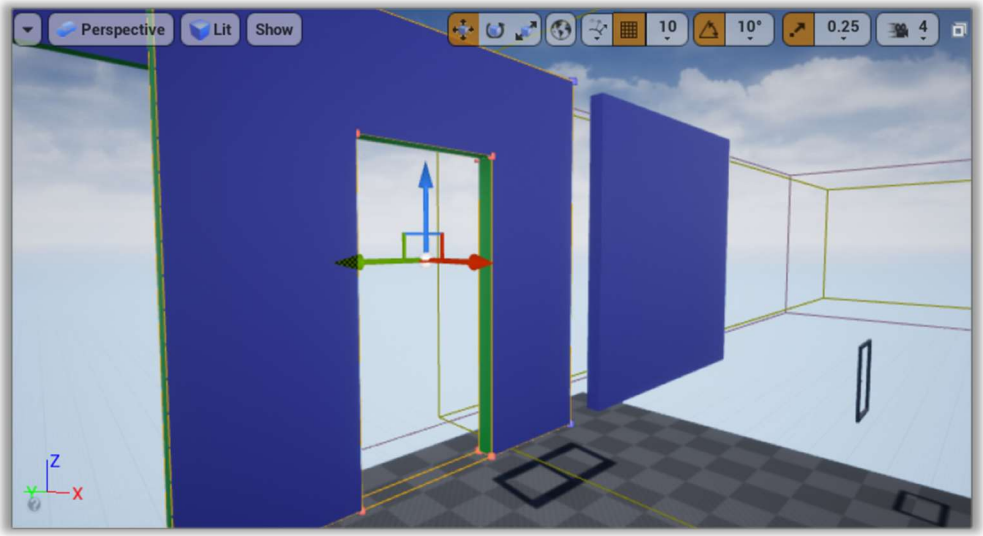


Ilustración 53 - Creación de pieza pared puerta desde pared

En este paso es importante tener en cuenta las medidas de nuestro personaje y su malla de colisión ya que en caso de ser medidas inferiores, no podremos atravesar el hueco de puerta (siempre que la colisión entre objetos esté habilitada, como en nuestro caso).
Por otro lado, la pared puerta, contará con otro objeto adicional, la puerta.
Este objeto no ha sido creado desde cero en su totalidad.

Desde la propia web de Evermotion [69] podemos obtener el paquete "Archviz Vol.1" donde podremos encontrar modelos de arquitectura completos.

En nuestro caso, hemos utilizado la puerta principal de la escena 4, que además viene en dos partes, una parte interior y otra exterior.

Para ajustar la puerta a nuestro hueco, hemos utilizado una caja de tipo rectangular, con las medidas del ancho de la pared, y restando además la puerta exterior y la puerta interior para evitar efectos de clipping en la pieza.

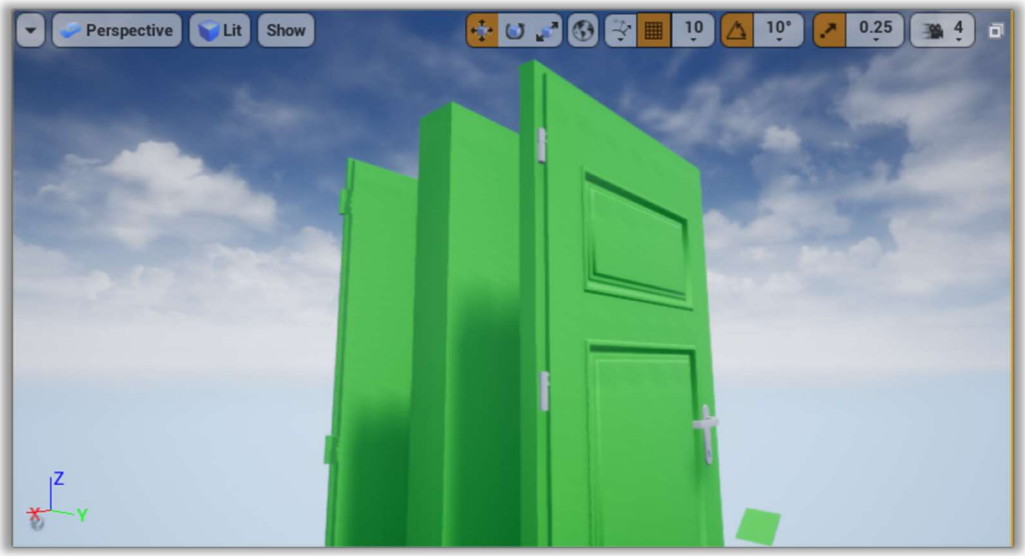


Ilustración 54 - Creación de puerta en tres piezas

UE4 tiene una peculiaridad, al menos hoy en día negativa y es que no se pueden reconfigurar los pivot points cuando importamos un objeto 3D o cuando convertimos un objeto en malla estática. Es decir, se pueden modificar sobre objetos que estén incluidos en la escena, pero no sobre el propio modelo en sí, por lo que es importante tener en cuenta este punto, y modificar los puntos de pivote de cada pieza antes de que los convirtamos en mallas estáticas.

Ahora que ya se dispone de las tres piezas que componen la puerta, hay que elegir una para convertirla en pieza padre, ya que sobre esta pieza será donde se modificará el pivote, y las otras dos piezas heredarán la posición de esta pieza padre al ser convertidas en una pieza única.

Ilustración 55 - Pieza padre e hijas

Una vez se ha comprobado con las piezas de puerta superpuestas en el hueco, y el pivote reajustado, que éstas rotan correctamente al hacer una rotación manual, es decir, que no se superponen con la pared y el movimiento tiene sentido, se puede convertir en una pieza única.

Es importante destacar que no se puede convertir el hueco de pared y la puerta en una única pieza. Debemos tratarlos como dos objetos independientes y será a nivel de blueprint donde formarán una única pieza compuesta por dos mallas estáticas. En caso de que convirtiésemos tanto la pared como la puerta en una única figura, no se podrá añadir después la animación de apertura/cierre.

El proceso utilizado es idéntico para el diseño de la pieza Pared Puerta Interior, sólo que, en este caso, los muros se realizan con una profundidad más estrecha, y se utiliza otro modelo distinto para la puerta. La pieza pared puerta aún tiene más peculiaridades, pero éstas son a nivel de blueprint que se comentarán a lo largo del desarrollo del presente capítulo.

- **Pared Ventana:** Para construir la pieza de pared que lleva incluida una ventana, se utiliza como referencia el modelo de pared normal. Posteriormente desde el paquete utilizado para conseguir la puerta, "Archviz Vol.1" de Evermotion [69], buscamos algo que se adapte como marco de ventana para nuestro modelo y el diseño final que queremos utilizar para nuestro videojuego.

  Una vez que ya se ha encontrado la pieza que se desea adaptar, mediante sustracción, se elimina de la parte de la pared el hueco donde se va a colocar nuestra ventana, y una vez hecho, se tendrá que crear dos piezas más que serán las que nos servirán para generar los cristales y por tanto la transparencia y visibilidad desde dentro y fuera de la casa.

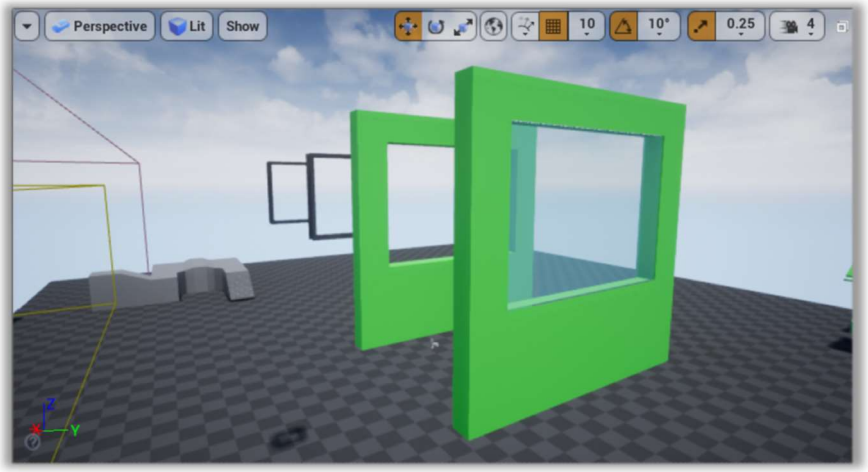


Ilustración 56 - Despiezado de la pieza Pared Ventana

Al igual que en el resto de los procesos, una vez ya se disponga de todo, se tendrá que colocar jerárquicamente dependiendo de un padre, para poder asignar correctamente el punto de pivote a la figura para posteriormente ser convertido en una malla estática.

- **Escalera:** La creación de la escalera fue quizá una de las piezas más complejas ya que había que tener en cuenta varios puntos, para que posteriormente todo pudiese encajar a la perfección. De hecho, en fases posteriores del proyecto snap-in y testeo la escalera se tuvo que rehacer de cero en numerosas ocasiones, forzando que además otras piezas también tuvieran que rehacerse con nuevas dimensiones acordes a las directrices de la escalera.

  En la sección de geometría de UE4, éste incorpora distintos modelos de geometría de escalera simples.

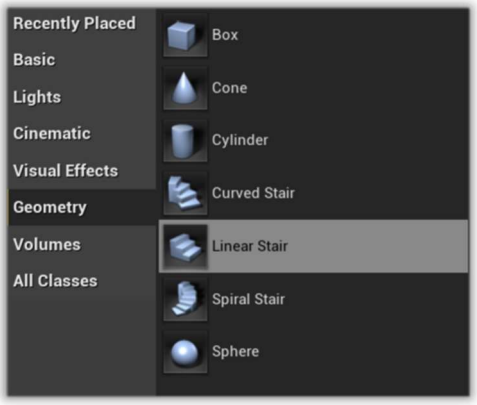


Ilustración 57 - Tipos de escalera disponibles

En nuestro caso, se ha decidido utilizar el tipo Linear Stair.

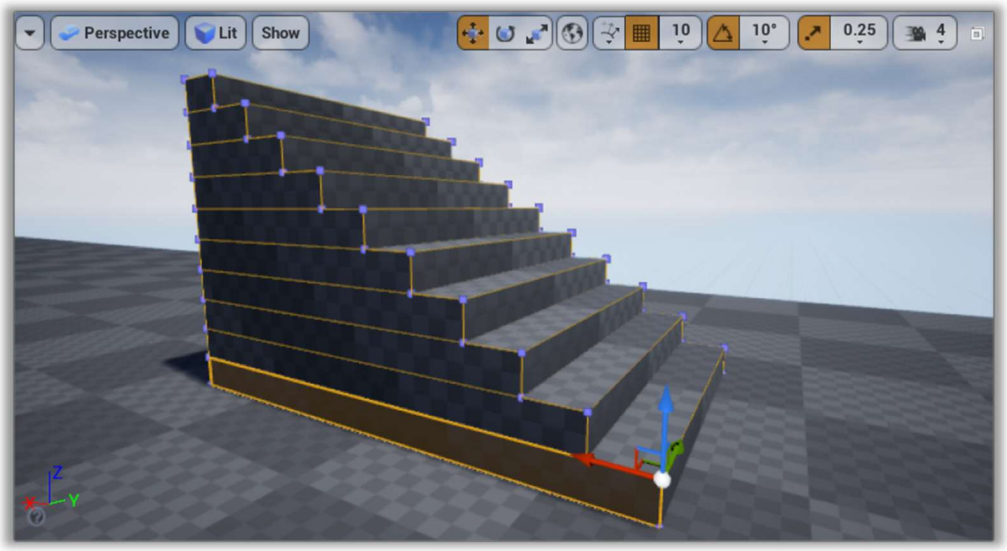


Ilustración 58 - Diseño de escalera mediante geometría

El principal problema de la escalera reside en la simetría y uniformidad que se buscaba a la hora de realizar este componente. Se deseaba que tuviese los mismos escalones tanto en la parte media-inferior como en la parte media-superior. Además de eso, la altura de los escalones tenía que guardar cierto sentido, para que nuestro personaje pudiese subir por ellas sin problemas y que cuando la escalera conectase con un techo (como comentamos anteriormente, suelo de la planta superior) no hubiese problemas de paso por la altura del propio personaje (donde en numerosas ocasiones, el personaje no podía pasar correctamente) y el descanso de escalera, tenía que ser lo suficientemente ancho, como para que el personaje pudiese parar en él sin dar una sensación de estar encajonado.

Para el ajuste de la escalera, se utilizaban piezas de ejemplo colocadas en nuestro mapa, simulando un caso real, es decir, se colocaba un suelo y 4 paredes alrededor para ir regulando además de un techo que conectase con la escalera.

Una vez que (por fin) se consigue ajustar todo correctamente al milímetro, se añade nueva geometría de sustracción para dar a la escalera el diseño que se buscaba desde un primer momento, llegando al resultado de la siguiente imagen.

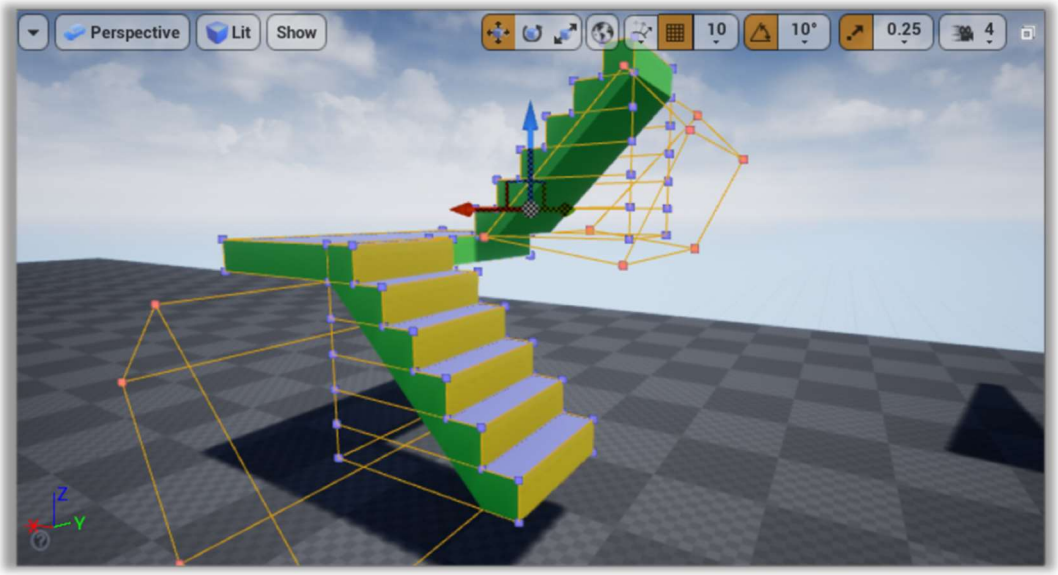


Ilustración 59 - Diseño final de escalera

- **Tejado/Teja/Teja Horizontal:** Aunque se trate de piezas diferentes, todas ellas guardan relación entre sí.

  Una vez que el jugador ya tenga la casa perfectamente construida, había que buscar una forma de cerrar la estructura, y aunque hay otras opciones que se incluyen en el juego, el diseño del tejado fue considerado un punto imprescindible.

  Aquí había que tener en consideración varios puntos.

Ilustración 60 - Modelos de colocación tejas / Tejado

Cuando se tienen dos piezas, el cierre del tejado es sencillo mediante sus tejas ya que formará un cierre regular. Sin embargo, cuando queremos ampliar y hacer el ancho de la casa de más piezas, el cierre del tejado quedará incompleto. Por ello se crearon las piezas llamadas “Teja horizontal” donde

las tejas laterales cerrarán en su parte posterior con un tejado, y las piezas centrales lo harán con techos.

- **Valla:** Se quería construir un cercado bien diseñado para la casa, utilizando las propias herramientas que nos ofrece Unreal Engine para el diseño de geometría. Para ello, se tomó como referencia el ancho de la pieza del suelo.

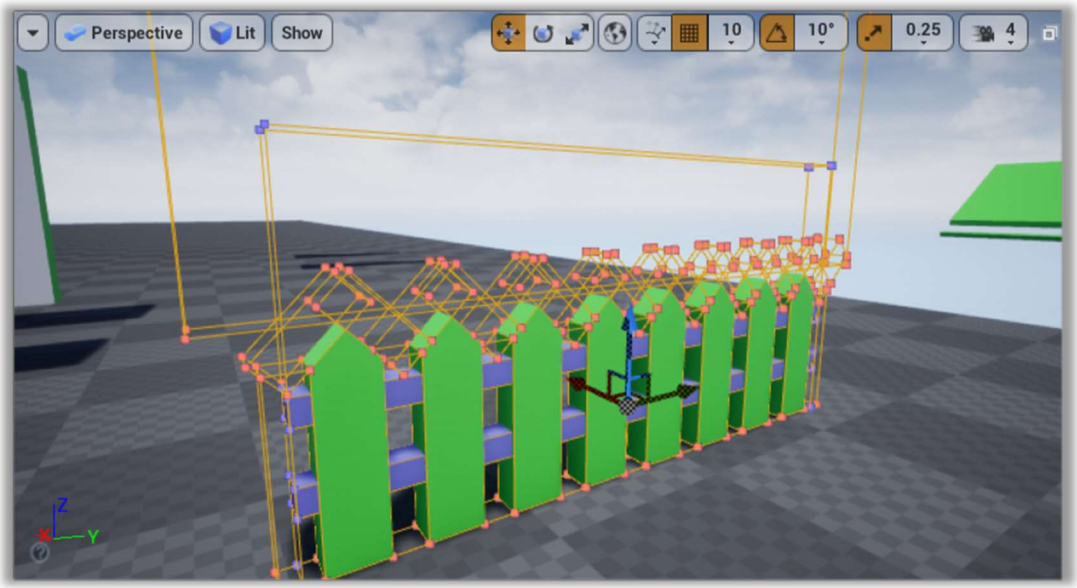


Ilustración 61 - Proceso de creación de la pieza valla

De forma laboriosa y precisa, mediante adición y sustracción se consiguió elaborar una valla perfecta y simétrica que pueda servir para el cercado de la casa.

- **Resto de piezas:** Ya hemos destacado los procesos de creación para las piezas más complejas durante su desarrollo. El proceso de creación sigue las mismas directrices de los ya mencionados anteriormente para las demás. A continuación mencionaremos el resto de las piezas incluidas en juego como base para la creación de estructuras.

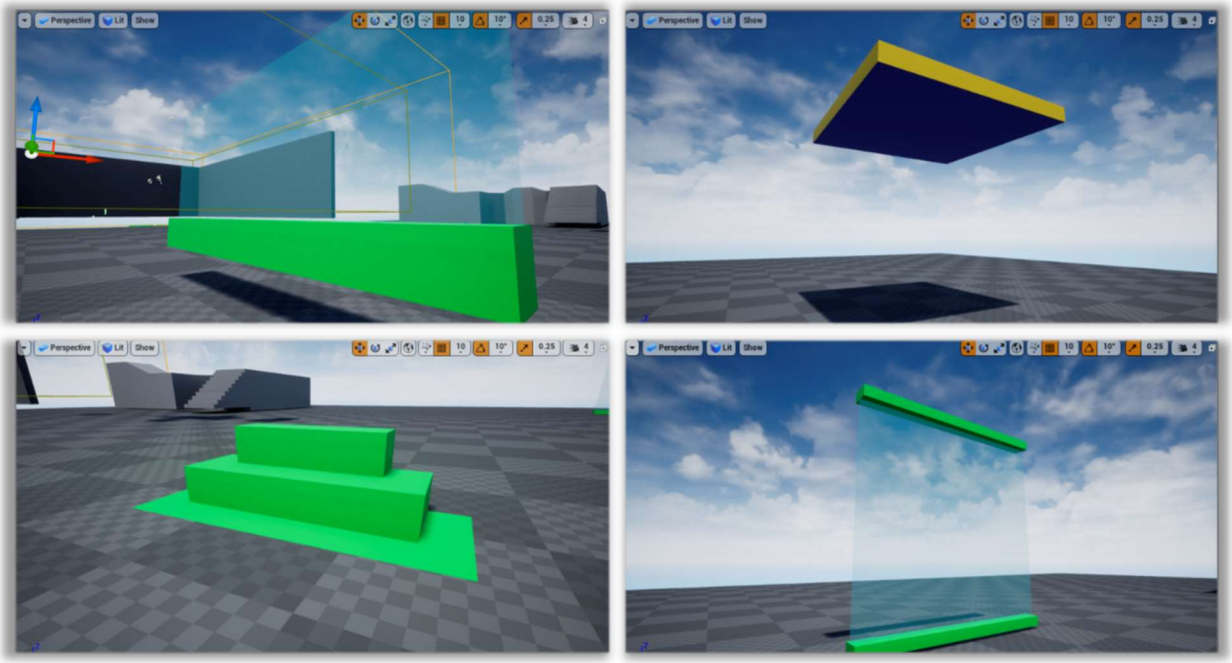


Ilustración 62 - Resto de piezas para utilizar en la construcción

4.5.2. Fase 2 – Incluir las piezas de forma interactiva en el juego

Una vez que ya tenemos todas las piezas creadas y definidas y asociadas a un BluePrint (recordemos que es necesario haber asociado las mallas estáticas a un actor que podamos incluir en el juego) es necesario crear algún tipo de interacción para poder incluirlas en juego.

Para ello, se crea un Widget Blueprint y se asignan los diferentes botones para cada una de las piezas que posteriormente desencadenarían en otras funciones tal y como se indica en el RF-3

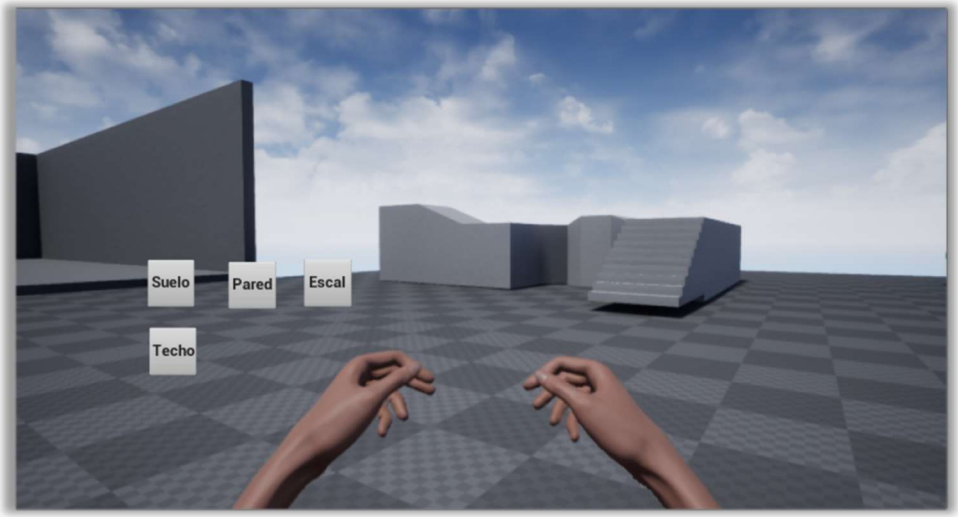


Ilustración 63 - Botones generados manualmente

A los inicios del desarrollo de Home Factory, estos botones eran **creados a mano**, uno a uno en la interfaz de usuario, por cada una de las piezas para verificar su correcto funcionamiento.

En fases posteriores y de cara a optimizar y mejorar el código del videojuego, aun siendo ya funcional, se empezó a trabajar con "Data Tables".

Las Data Tables son como una especie de formulario donde introducimos datos de entrada sobre aquello que queramos tratar. Con una estructura similar a la comúnmente conocida aplicación de escritorio, Excel [70] de Microsoft, las Data Tables no son más que estructuras CSV (Comma Separated Values).

	Row Name	Nombre	BluePrint_Asociado	Tipo Objeto	Precio	Valor	SM_1
1	Suelo	Suelo	BlueprintGeneratedClass'/Game/Female_Hands/Blueprints/BP_SueloChi	Suelo	100.000000	10.000000	StaticMesh'/Game/Meshes/SM_Foundation320.SM_Foundation320'
2	Pared	Pared	BlueprintGeneratedClass'/Game/Female_Hands/Blueprints/BP_ParedChi	Pared	100.000000	10.000000	StaticMesh'/Game/Meshes/SM_Pared320_2.SM_Pared320_2'
3	Techo	Techo	BlueprintGeneratedClass'/Game/Female_Hands/Blueprints/BP_TechoCh	Techo	100.000000	10.000000	StaticMesh'/Game/Meshes/SM_Techo.SM_Techo'
4	Escalera	Escalera	BlueprintGeneratedClass'/Game/Female_Hands/Blueprints/BP_Escalera(	Escalera	100.000000	10.000000	StaticMesh'/Game/Meshes/SM_Escalera320.SM_Escalera320'
5	ParedPuerta	ParedPuerta	BlueprintGeneratedClass'/Game/Female_Hands/Blueprints/BP_P_ParedF	Puerta	100.000000	10.000000	StaticMesh'/Game/Meshes/ParedPuerta.ParedPuerta'
6	Teja	Teja	BlueprintGeneratedClass'/Game/Female_Hands/Blueprints/BP_TejaChilc	TejaIzquierda	0.000000	0.000000	StaticMesh'/Game/TejaIDer.TejaIDer'
7	Tejado	Tejado	BlueprintGeneratedClass'/Game/Female_Hands/Blueprints/BP_TejadoCl	Tejado	0.000000	0.000000	StaticMesh'/Game/Meshes/Teja7.Teja7'
8	Valla	Valla	BlueprintGeneratedClass'/Game/Female_Hands/Blueprints/BP_VallaChil	Suelo	0.000000	0.000000	StaticMesh'/Game/Meshes/Valla.Valla'
9	TejaHorizontal	TejaHorizontal	BlueprintGeneratedClass'/Game/Female_Hands/Blueprints/BP_TejaHoriz	Suelo	0.000000	0.000000	StaticMesh'/Game/Meshes/TejaHorizontal2.TejaHorizontal2'
10	EscaleraEntrada	EscaleraEntrada	BlueprintGeneratedClass'/Game/Female_Hands/Blueprints/BP_Escaleras	Suelo	0.000000	0.000000	StaticMesh'/Game/Meshes/Esc_EntradaV2.Esc_EntradaV2'
11	ParedCristalVentanas	ParedCristalVentanas	BlueprintGeneratedClass'/Game/Female_Hands/Blueprints/BP_P_Pared\	Suelo	0.000000	0.000000	StaticMesh'/Game/Meshes/ParedCristalVentanas3.ParedCristalVentana
12	Barandilla	Barandilla	BlueprintGeneratedClass'/Game/Female_Hands/Blueprints/BP_Barandill	Suelo	0.000000	0.000000	StaticMesh'/Game/Meshes/Barandilla2.Barandilla2'
13	ParedCristal	ParedCristal	BlueprintGeneratedClass'/Game/Female_Hands/Blueprints/BP_P_Pared(	Suelo	0.000000	0.000000	StaticMesh'/Game/Meshes/ParedAcristalada.ParedAcristalada'
14	ParedInterior	ParedInterior	BlueprintGeneratedClass'/Game/Female_Hands/Blueprints/BP_P_Paredl	Suelo	0.000000	0.000000	StaticMesh'/Game/Meshes/ParedInterior2.ParedInterior2'
15	ParedPuertaInterior	ParedPuertaInterior	BlueprintGeneratedClass'/Game/Female_Hands/Blueprints/BP_P_ParedF	Suelo	0.000000	0.000000	StaticMesh'/Game/Meshes/ParedPuertaInterior.ParedPuertaInterior'
16	TejaDer	TejaDer	BlueprintGeneratedClass'/Game/Female_Hands/Blueprints/BP_TejaChilc	Suelo	0.000000	0.000000	StaticMesh'/Game/TejaDizq2.TejaDizq2'
17	Locked1	Locked1	BlueprintGeneratedClass'/Game/Female_Hands/BP_Candado.BP_Canda	Suelo	0.000000	0.000000	StaticMesh'/Game/Female_Hands/16884_simple_padlock_v1.16884_sim

Ilustración 64 - Tabla Datos Construcción

La razón por la que se decidió utilizar estas tablas reside en la facilidad para cambiar datos de cada una de las piezas, desde un panel unificado, sin tener que realizar estos cambios de forma individualizada. Dentro de esta tabla, se tendrán las referencias a los nombres, BluePrint a utilizar, el tipo de objeto, precio, valor, malla estática asociada, etc.

Una vez se haya implementado este sistema, sólo se tendría que realizar una consulta, sobre todos los ítems que estuviesen incluidos en esta tabla, para generar tantos botones como ítems estén incluidos en ella.

Se necesitan crear dos "Widget Blueprint", uno se llamará "de interfaz", y servirá como estructura base para recoger toda la información que aparecerá en pantalla para el menú de los objetos. Como los botones van a ser dinámicos, es decir, se van a generar uno a uno según los datos que recojamos de la "Data Table" se necesitará además otro widget blueprint, que se llamará de botón, para definir cómo serán estos botones.

Nuestro Widget Blueprint de interfaz, para esta etapa de pruebas, será bastante sencillo ya que sólo necesitamos incorporar una caja de ítems verticales, cuyos ítems se recibirán de la llamada a la tabla donde tenemos almacenados todas las piezas; y por cada una creará un widget blueprint de botón, que será añadido como un hijo de esta lista vertical.

De esta forma, cada vez que modifiquemos Data Table, esta lista también será actualizada.

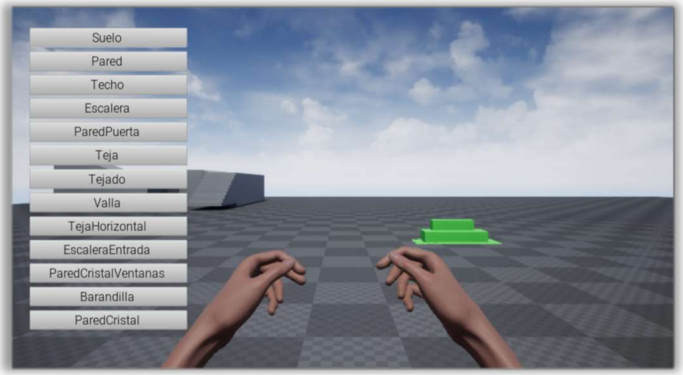


Ilustración 65 - Botones generados a partir de la tabla de forma automática

4.5.3. Fase 3 – Clase BP_Objeto

Esta clase es una de las más importantes que componen Home Factory.

Cada vez que se llama a esta clase, se crea un objeto en el escenario, que no representa nada, ya que no tiene ninguna figura asociada, sin embargo, dispone de una función "ActualizarObjeto" que recibirá los datos que hayan sido seleccionados desde el menú de selección de pieza, y haciendo una llamada a la tabla de datos mencionada en el apartado anterior con esta información, la malla estática de BP_Objeto, mutará a la malla estática de la selección.

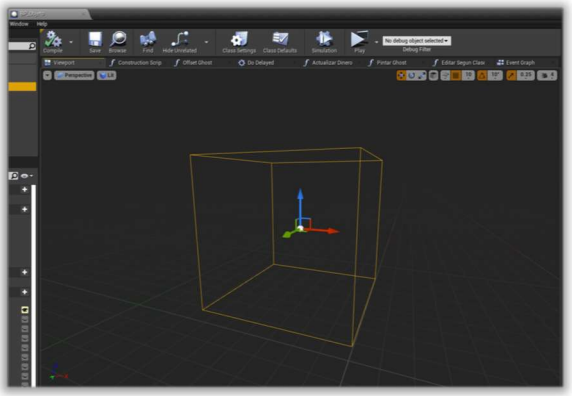


Ilustración 66 - Caja vacía de BP_Objeto lista para mutar

Una vez que el objeto ha mutado, ahora se puede visualizar delante del jugador como lo que se denomina "Ghost", que no es más que una pieza fantasma, que aún no ha sido colocada en el escenario y que mediante una interpretación de colores que se asignará posteriormente (rojo, amarillo, verde) el jugador podrá comprobar la viabilidad de colocar esa pieza donde esté apuntando.

4.5.4. Fase 4 – Uso de LineTrace

Ahora que ya se dispone de la base diseñada, se necesita crear una relación entre nuestro personaje, donde está apuntando el jugador (requisito RF-12), el objeto ghost y la ubicación final de la pieza en el escenario.

Unreal Engine dispone de una función propia llamada LineTraceByChannel que proyecta una línea según los parámetros de entrada que demos a la función (vector delantero y coordenadas del mundo de la cámara) y nos devolverá multitud de información en lo que se denomina "hit de trazado": Saber si la línea está haciendo hit con algo o no, distancia, posición de hit, actor al que apunta, componente al que apunta, etc.

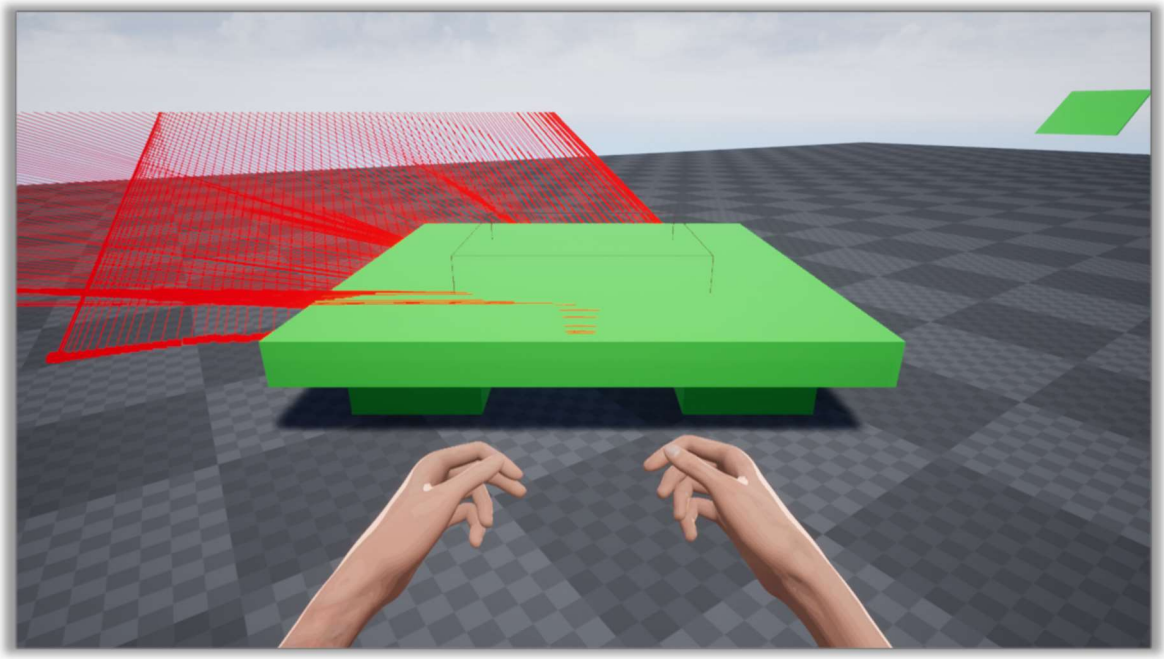


Ilustración 67 – Muestra de la proyección de LineTrace

De esta forma se envía la información a nuestro ghost. Si la línea de trazado no golpea con nada, significa que el jugador está apuntando con la pieza al aire, es decir,

la posición de la pieza será la posición de la cámara multiplicada por la distancia a la que queramos alejarla.

Si la línea de trazado está golpeando con algo significa que el jugador quiere colocar esa pieza, por tanto se tiene que recoger las coordenadas donde se está apuntando y enviarlas al ghost, para que éste se coloque de acuerdo la parte del escenario a la que apunta.

4.5.5. Fase 5 – Snap-In

Ahora que ya somos capaces de colocar las piezas sobre el escenario hay que añadir un poco de lógica a la colocación de las mismas. ¿Se desea que las piezas se coloquen libremente? O por el contrario ¿se desea que las piezas se autoajusten entre sí? Para Home Factory, se terminó decidiendo por la implementación de un sistema de Snap-in o autoajustado entre piezas según se indica en el requisito RF-6. De esta forma el resultado final una vez creada la estructura resultaría ser mucho más preciso, simétrico y elegante.

Primero había que tener en cuenta una tabla de interacción ya que no todas las piezas deben auto-ajustarse entre sí al igual que no todas pueden ser colocadas unas sobre otras según se hace referencia en el requisito RF-5, por ejemplo, no tiene sentido ajustar un suelo con un techo, pero sí lo tiene ajustar una pared con un suelo.

	Suelo	Pared	Techo	Escalera	ParedPuerta	Teja	Tejado	Valla	TejaHorizontal	EscaleraEntrada	ParedCristalVentanas	Barandilla	ParedCristal
Suelo	x												
Pared	x	x	x		x						x		x
Techo		x	x		x						x		x
Escalera	x		x										
ParedPuerta		x			x						x		x
Teja		x			x						x		x
Tejado						x	x						
Valla	x							x					
TejaHorizontal		x			x						x		x
EscaleraEntrada					x								
ParedCristalVentanas		x	x		x						x		x
Barandilla			x										
ParedCristal		x	x		x						x		x

Ilustración 68 - Tabla de interacción de autoajustes

Una vez que se tiene clara la interacción necesaria, se empieza a diseñar la función asociada a cada uno de estos autoajustes. Por ejemplo, para el ajuste automático de los suelos, se toma como referencia la posición de la figura a la que el jugador está apuntando y mediante comparación de posiciones y el valor de los signos, determinamos dónde será la posición más acertada para la figura que se va a colocar.

Por otro lado, hay que tener en cuenta que hay muchos métodos de autoajuste que no tienen por qué ser duplicados, sino reutilizados, es decir, un ajuste entre una pared y un suelo, no se podrá utilizar para ajustar dos paredes, pero sí se podrá utilizar para una paredinterior y un suelo, de la misma forma que el ajuste de dos paredes podrá ser reutilizado para una paredventana.

4.5.6. Fase 6 – Colocación de piezas, actores colocados y contadores

Como requisito para la colocación de piezas, se incorpora el requisito de existencia de una pieza de suelo mínima para poder continuar con la creación de estructura, tal y como se indica en el RF-4.

Una vez que las piezas ya pueden ser colocadas y son autoajustadas, se tiene que almacenar toda esta información para ser utilizada en algún otro momento. En concreto se necesitará llevar este control para las funciones de guardar/cargar partida.

Actores colocados es un array que recoge los valores de la pieza que se están colocando en ese momento (su tipo, su posición, su rotación, etc.). También se tendrá en cuenta las texturas que están aplicadas a cada una de las piezas de actores colocados y la altura de las piezas.

La altura de las piezas no indica la posición del mundo sobre las piezas en concreto, sino que se considera altura como “pisos” sobre los que las piezas forman parte. De esta forma, un suelo, una pared y un techo, formarán parte de la “altura 0” mientras que, si se coloca una pared de un nivel superior sobre ese techo, esa pared formará parte de las piezas de “altura 1” y así sucesivamente.

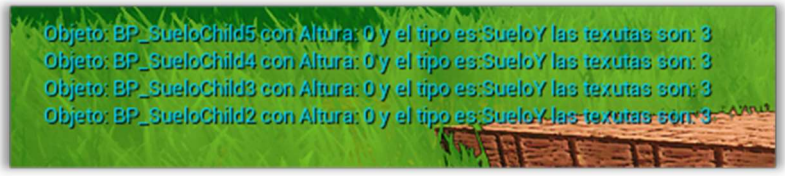


Ilustración 69 - Control de actores y texturas

Por otro lado, además deberá tenerse en cuenta el coste de cada una de las piezas que estamos colocando y el valor que supone cada una de estas, para modificar ambos contadores en partida.

4.5.7. Fase 7 – Rotación e intercambio de piezas

Las piezas inicialmente serán colocadas de forma perpendicular al jugador, por ello queríamos implementar un sistema donde fuese posible rotar dichas piezas. Hay que tener en cuenta que esta función sólo será útil para los suelos ya que el resto de las piezas serán autoajustadas, luego será el suelo quien determinará la rotación de las piezas y de la construcción al completo.

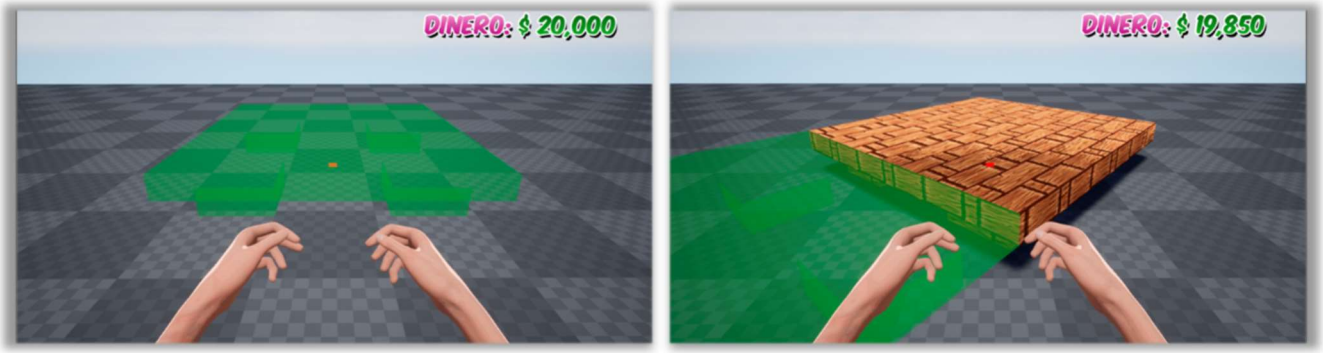


Ilustración 70 - Ejemplo de rotación

Dicha rotación se podrá ejecutar moviendo la rueda del ratón hacia arriba o abajo para girar la pieza en sentido horario o antihorario.

La función de rotación en algunos casos no funciona de esta forma si las piezas están siendo ya autoajustadas. Por ejemplo, en el caso de la escalera, una vez que ésta está siendo autoajustada con el suelo, mover la rueda del ratón rotará la pieza 90º.

Otras piezas, sin embargo, como las tejas, para no ser duplicadas (ya que para esta pieza había que invertir las partes editables) se intercambian en tiempo de ejecución según rotemos la pieza, de forma que el jugador es capaz de ver que la pieza está cambiando su dirección cuando en realidad lo que está sucediendo es un intercambio de blueprints.

Lo mismo sucede para las piezas ParedInterior y ParedPuertaInterior. En este caso no necesitamos realizar ningún tipo de rotación, sin embargo, cuando el jugador esté intentando colocar alguna de las piezas de tipo pared entre dos suelos, éstas serán directamente intercambiadas por ParedInterior y por ParedPuertaInterior en caso de estar realizando esta acción con una ParedPuerta.

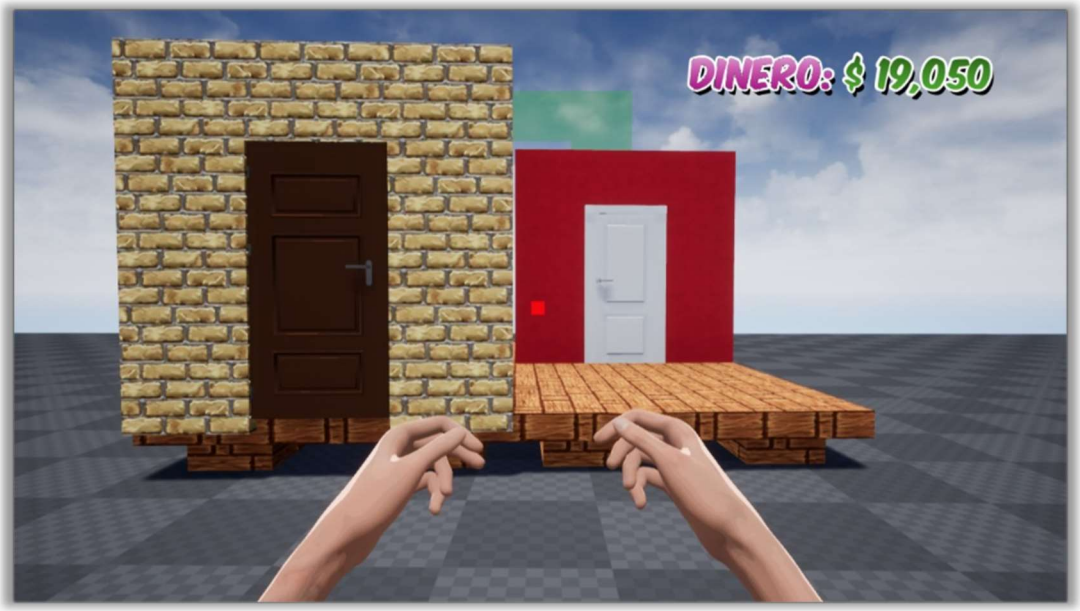


Ilustración 71 - ParedPuerta y ParedPuertaInterior

Además de evitar añadir piezas similares en el catálogo de piezas, lo que se pretendía evitar principalmente con esta implementación es que un jugador fuese capaz de añadir una pieza del tipo ParedInterior ó ParedPuertaInterior en el exterior de la casa.

4.5.8. Fase 8 – Menú edición

Una vez que ya se dispone de las piezas para colocar en la partida, se deseaba dar al jugador la posibilidad de editar visualmente esas piezas proporcionándole un menú de selección de texturas, una previsualización sobre la pieza en cuestión como se indica en el requisito RF-16 y varias opciones de aplicar esas texturas (aplicando a una sola o a todas del mismo nivel al mismo tiempo) como se indica en el requisito RF-8.

Para proceder con este menú, además de la creación de interfaz gráfica, correspondiente había que tener en cuenta varios factores. En primer lugar, había que añadir un "box collision" para poder detectar si el jugador estaba apuntando a la parte interior o exterior de la pieza. Hay que pensar que no es lo mismo editar la parte exterior de una pared, que corresponderá con piedra, ladrillo, etc, que la parte interior de la pared que va asociada con la pintura de la habitación.

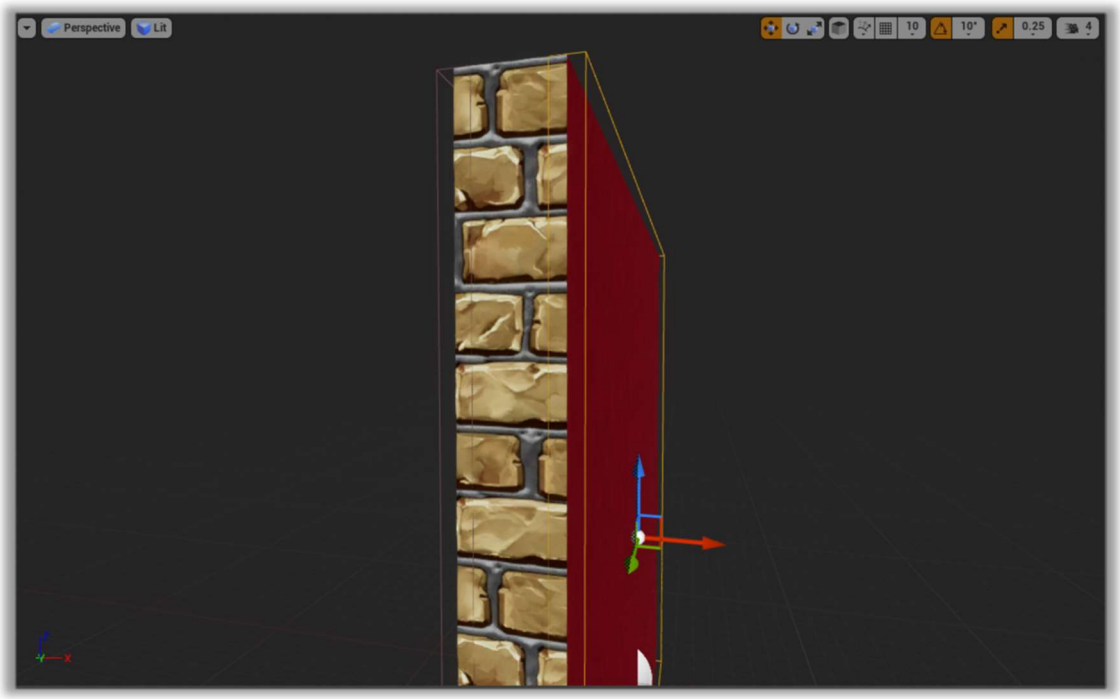


Ilustración 72 - Box collision interior y exterior en pared

Ahora que ya es posible identificar dónde estamos apuntando, hay que transferir este "índice de textura" a la función encargada de aplicar la textura seleccionada. De esta forma, cuando se seleccione la pared (por ejemplo) por su cara exterior sólo nos aparecerán los materiales asignados al array de materiales 1 (los de tipo exterior, como ladrillos y variantes) y si se selecciona por la cara interior, sólo nos aparecerán los materiales asignados al array de materiales 2 (los de tipo interior, como los distintos tipos de pintura)

Dentro del menú edición además y en función de la pieza a la que estemos apuntando, se incluye la posibilidad de aplicar esta textura a una sola pieza o a todas las del mismo tipo para no hacer de esta acción de edición algo aburrido y cansado para el jugador.

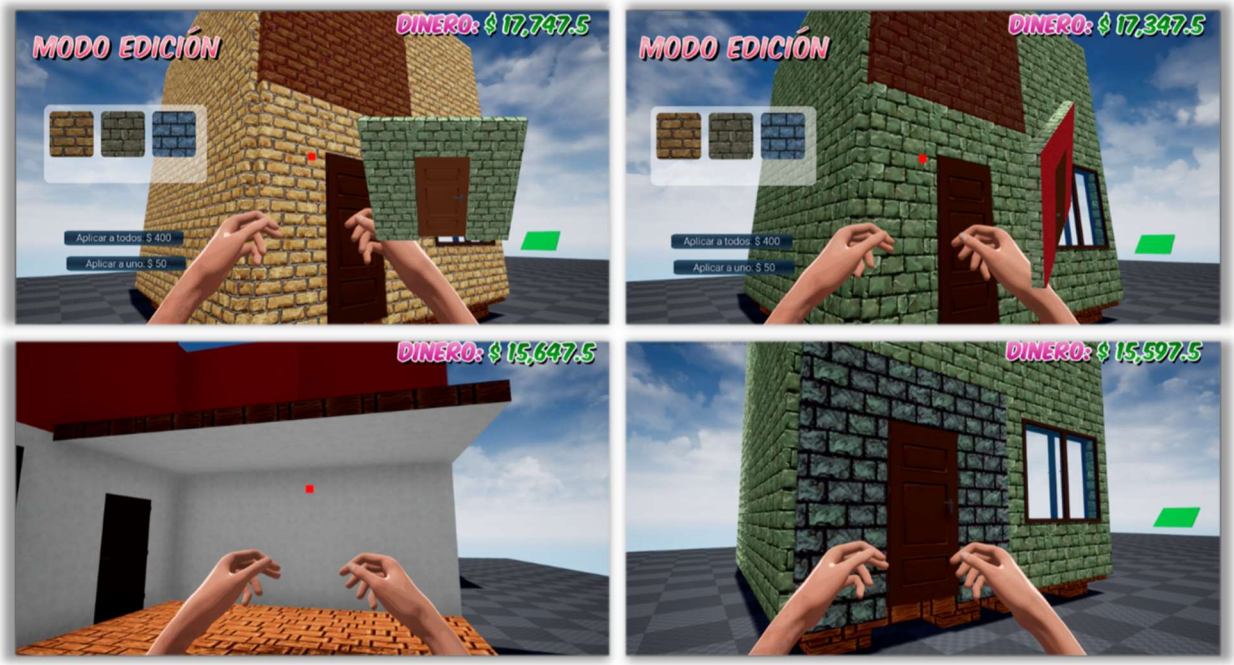


Ilustración 73 - Diversos modos de edición de texturas

Por ejemplo, si se selecciona una pared desde fuera, podremos modificar la textura exterior de todas las piezas de tipo pared, sin embargo si se hace con una pared desde dentro, la opción aplicar a todas, servirá para modificar el color de pintura de todas aquellas piezas de tipo pared que estén a la misma altura con un solo clic.

4.5.9. Fase 9 – Reembolso de piezas

Llegados a este punto, hay que empezar a considerar ciertos detalles. ¿Qué pasa si el jugador se equivoca colocando una pieza que no quería colocar, ha colocado por error o se ha cerrado en una habitación con cuatro paredes sin una puerta?

En la mayoría de los videojuegos, el jugador debe tener algún tipo de "castigo" o penalización por sus errores o malas decisiones. En este caso, también se ha considerado este punto como se indica en el requisito RF-19, pero ofreciendo la opción de corrección de error, es decir, un jugador podrá efectuar un reembolso de las piezas colocadas, sólo que en lugar de no recibir ningún tipo de reembolso por ellas (lo que consideraríamos una penalización del 100%), el jugador recibirá en su cuenta el 60% del importe de venta de la pieza colocada.

Es importante mencionar que el jugador no recibirá ningún tipo de importe asociado a la inversión en la edición de texturas de la pieza.

4.5.10. Fase 10 – Modificar aspecto visual de Home Factory

En esta fase, el videojuego ya resulta prácticamente funcional según los requisitos funcionales y no funcionales que habíamos determinado al inicio, sin embargo, aún hay algo muy importante en lo que tenemos que trabajar, el aspecto visual del título.

Se decide implementar asset de Stylized Forest [71] de la propia tienda de Epic Games para el escenario:

Ilustración 74 - Stylized Forest

Se buscaba un entorno que, como se ha mencionado ya en otras ocasiones, transmitiese al jugador cierta tranquilidad y libertad, y qué mejor para ello que un bosque repleto de árboles, colorido, con un lago.

Para las texturas de las piezas, se han utilizado las texturas del pack de texturas Stylized Texture Pack – Vol.05 [72]:

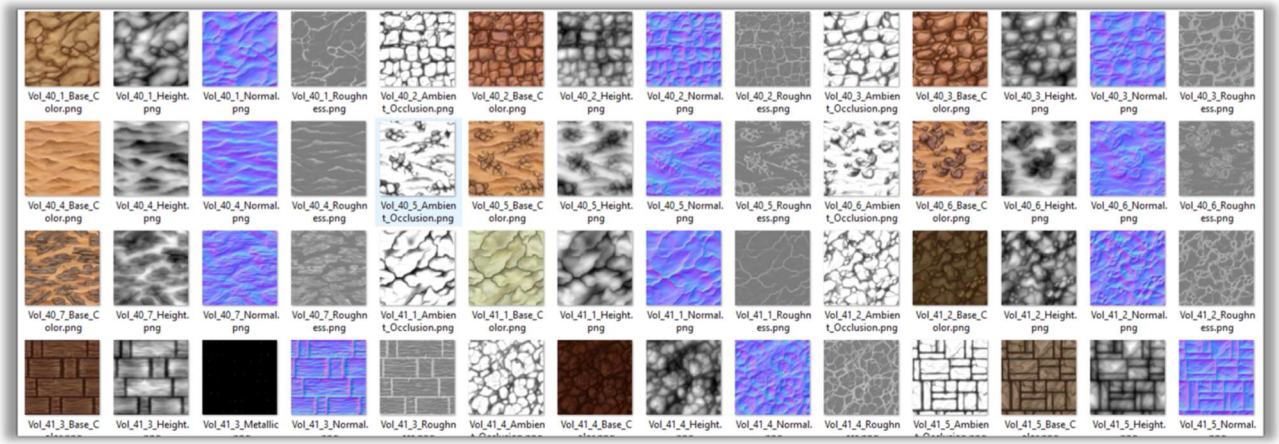


Ilustración 75 - Stylized Texture Pack Vol.5 LowlyPoly

Para añadir ese efecto Cel Shading a todas las texturas del título, se han utilizado las bases de materiales del pack Cartoon Cel Shader [73]:

Ilustración 76 - Cartoon Cel Shader

Para dar un poco más de perspectiva al título, se han incorporado unos brazos junto con la cámara en primera persona con el asset de Female Hands [74]. Además, las texturas de este assets han sido modificadas

Ilustración 77 - Female Hands

Para dar algún efecto más visual e interactivo, se decidió agregar a la colocación de las piezas (evento construct) una pequeña animación de timeline de escala de crecimiento como se describe en el requisito RF-7. Además, se queríamos añadir

algún otro elemento visual en la colocación, por ello se agregó este pack de partículas en la colocación de piezas llamado Stylized VFX Pack [75].

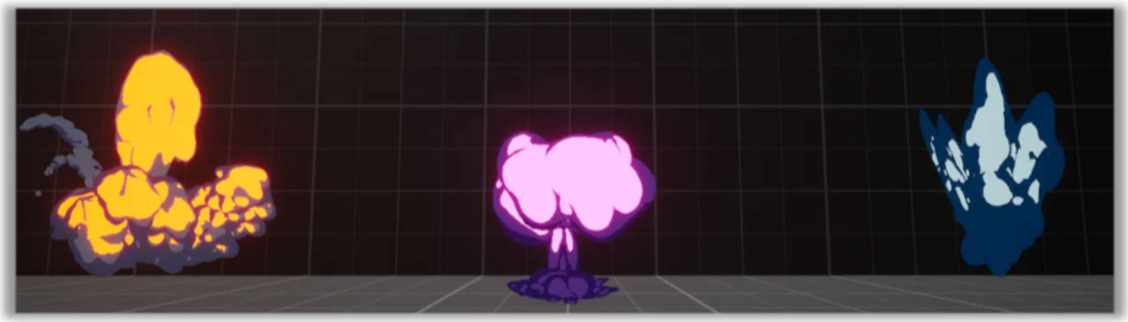


Ilustración 78 - Stylized VFX Pack

Para todos los assets implementados, se ha revisado el MDA (Marketplace Distribution Agreement) de Unreal Engine donde cita textualmente en el siguiente enlace:

https://www.unrealengine.com/en-US/marketplace-distribution-agreement

b. You agree that Epic may grant to Customers a non-exclusive, worldwide, and perpetual license to download, use, copy, post, modify, promote, license, sell, publicly perform, publicly display, digitally perform, distribute, or transmit the Content for personal, promotional, and/or commercial purposes ("**Digital Rights**"). Both Parties expressly acknowledge that distribution of the Content to Customers via the Marketplace is not a sale of the Content but the grant of Digital Rights to Customers. Such Digital Rights shall be granted pursuant to Epic's then-current end user license agreement for the Marketplace ("**EULA**").

Para la música tanto del menú principal como del bosque, se han utilizado las siguientes canciones desde YouTube:

- Butter-Fly (Full ver.) – Digimon Adventure OP [Piano] [76]
- Zelda Medley – Lindsey Stirling [77]

Combinadas con el siguiente efecto de fondo en ambos casos:

- GOOD MORNING SPRING NATURE THERAPY🌼ASMR Ambience Meditation🌼4k Flowery Meadow Birds Sounds for sleep [78]

4.5.11. Fase 11 – Guardar/Cargar Partida

Se deseaba implementar un sistema de guardado/cargado como se indica en el requisito RF-13 para que el jugador pueda guardar el estado actual de la partida y poder continuar exactamente en el mismo punto donde lo dejó.

La función de guardado se configura de la siguiente forma:

Se genera un SaveGame Object, y dentro de él se almacenan las variables que posteriormente vamos a necesitar cargar: estado de actores colocados en el mapa, posiciones de los actores, rotaciones, las texturas que hayan sido aplicadas, dinero del jugador, valor oculto de la casa, etc.

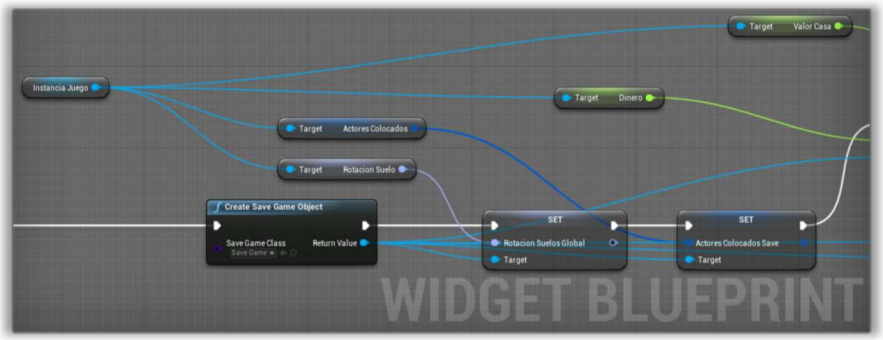


Ilustración 79 - Guardar partida

Para cargar partida, primero se debe comprobar si existe dentro de las partidas guardadas una partida que contenga el nombre de la partida que hemos creado.

Si existe, se carga y convierte al tipo "SaveGame Object" para obtener de él toda la información que se había guardado anteriormente. Toda esta información deberá ser enviada a nuestra InstanciaJuego ya que es allí donde almacenamos toda la información de tipo global entre clases.

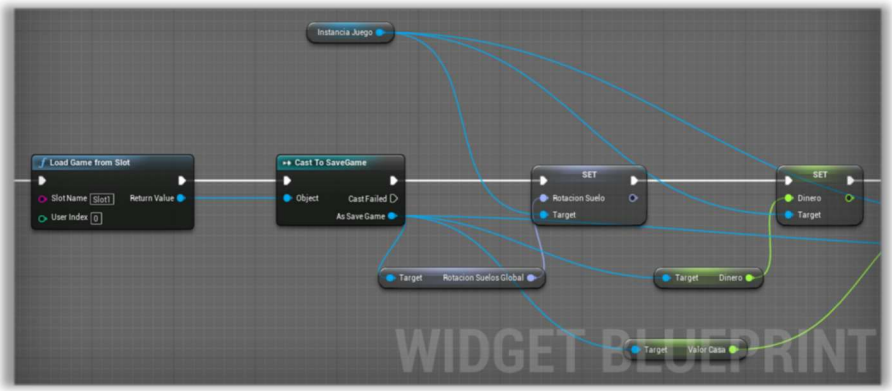


Ilustración 80 - Cargar partida

4.5.12. Fase 12 – Modo subasta

Otro de los factores que hacen diferente a Home Factory, es su sistema de subasta. En lugar de hacer del título un videojuego de supervivencia donde conseguir materiales y a partir de ahí poder elaborar las creaciones, se deseaba hacer del sistema de conseguir dinero, un elemento diferenciador.

Cada una de las piezas que se pueden colocar en el mapa disponen de un valor. Este valor se va almacenando en función de la estructura que el jugador haya construido. Una vez el jugador esté decidido a vender (o se haya quedado sin fondos para seguir construyendo) podrá entrar en el modo subasta. En el modo subasta se informará al jugador tanto de la inversión que ha realizado en la estructura como del valor que tiene la casa en ese momento. El jugador tendrá la opción de poder continuar con la subasta o salir.

Ilustración 81 - Modo subasta

Si el jugador acepta, aparecerán los posibles compradores con sus impresiones sobre la casa y posteriormente la puja de cada uno de ellos. Una vez se hayan mostrado todas las pujas el jugador podrá revisar el histórico de la conversación, aceptar la puja más alta que se ha realizado sobre la casa o rechazar la subasta.

Ilustración 82 - Aceptar/Rechazar subasta

Si el jugador opta por aceptar la subasta, a continuación, se mostrará una pantalla resumen indicando el dinero del que actualmente dispone, el dinero que ha ganado en la subasta y la suma de los dos mediante una animación en "dinero actual".

Ilustración 83 - Resumen de la venta

El algoritmo del modo subasta es bastante sencillo. Según el valor de la casa, se realizan dos multiplicaciones por 150 y 175 respectivamente. Dentro de ese rango, se escogen aleatoriamente números según el número de compradores (en esta versión de Home Factory, el número de compradores está limitado a 3) y se asigna un valor a cada uno de ellos. La puja máxima será el número más alto obtenido de entre los 3 números aleatorios.

Una vez el jugador acepta la subasta, se realiza un fundido a blanco, se reinicializa la variable del valor de la casa, se vacía el array de actores colocados y se vuelve a cargar nuevamente el mapa inicial. Esta vez el jugador tendrá más dinero para poder crear estructuras de mayor complejidad.

Capítulo 5 – Validación y pruebas de rendimiento

En este apartado se tratarán los resultados y distintas pruebas de rendimiento de la aplicación sobre la plataforma PC para la que ha sido diseñado.

5.1. Validación de resultados

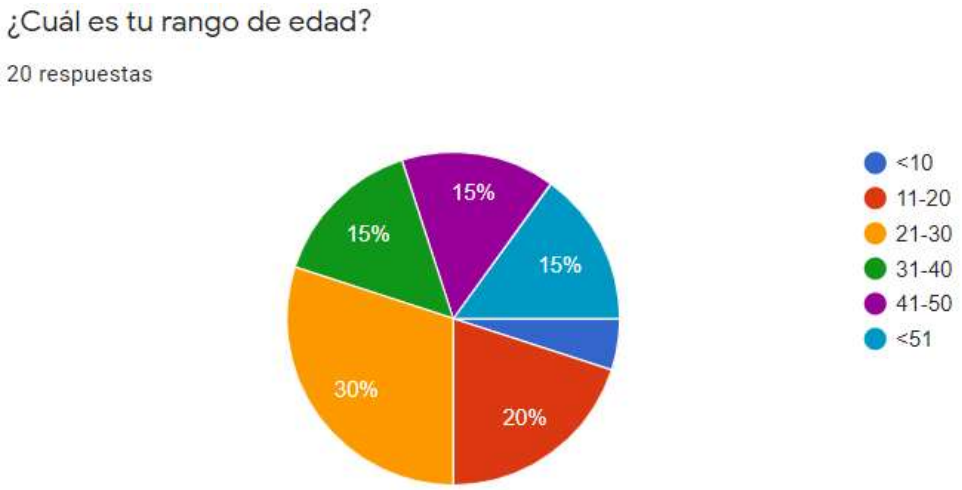
Antes de dar Home Factory por finalizado, se publica una versión de prueba del prototipo para que pueda ser testeada por diferentes personas desde sus propios equipos y además una encuesta creada en de Google Forms para así obtener información objetiva sobre Home Factory.

El formulario se puede comprobar mediante el siguiente enlace:

https://docs.google.com/forms/d/e/1FAIpQLSf6oj7oaIxpFWI0AmGC3LqO5kLLIsI93qTnCAKvBNYdTyBxYw/viewform

A continuación, vamos a analizar las respuestas recibidas de entre los 20 usuarios que han podido testear el juego y completar la encuesta después de hacerlo.

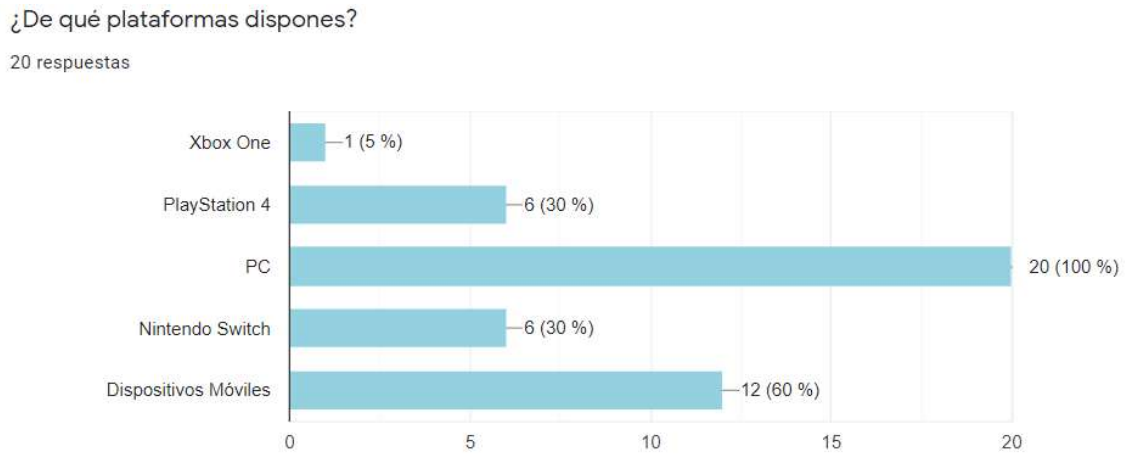
5.1.1. ¿Cuál es tu rango de edad?



De entre las 20 respuestas recibidas se puede encontrar que el rango más destacado de edad es de 21-30 años con un 30%, luego se puede entender qué tal encaja este género de construcción con una estética Cel Shading entre los jóvenes de 21-30 principalmente.

Para una primera toma de contacto, la encuesta se ha realizado con 20 usuarios aunque la idea es realizarlo en grupos mayores y tomar muestras más grandes para aumentar la precisión de los resultados.

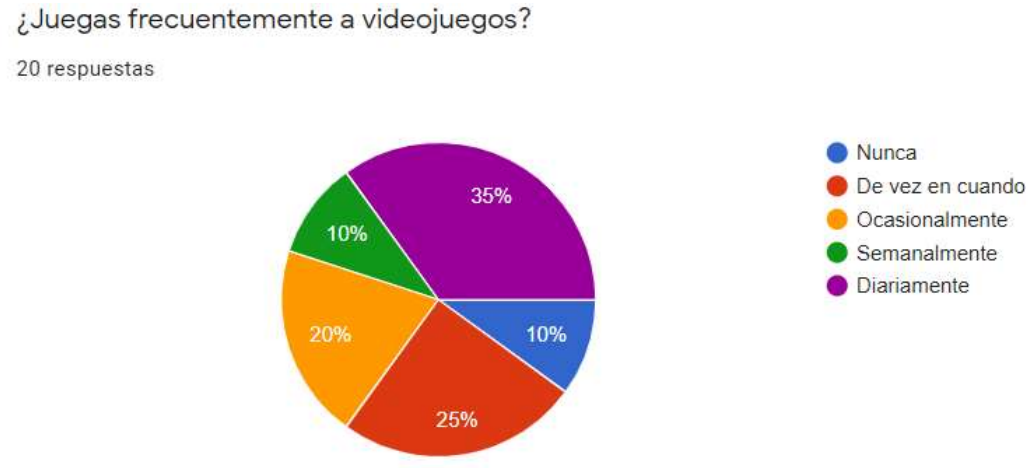
5.1.2. ¿De qué plataformas dispones?



Se puede comprobar como la mayoría de los usuarios, además de disponer de múltiples plataformas jugar, utilizan PC como plataforma para videojuegos.

Esto es importante ya que ayudará a entender la facilidad que ha tenido para el jugador entender el sistema de control del título. Por lo general, resultará más fácil para jugadores expertos en plataformas de PC que en jugadores que dediquen normalmente más tiempo a videojuegos en consolas.

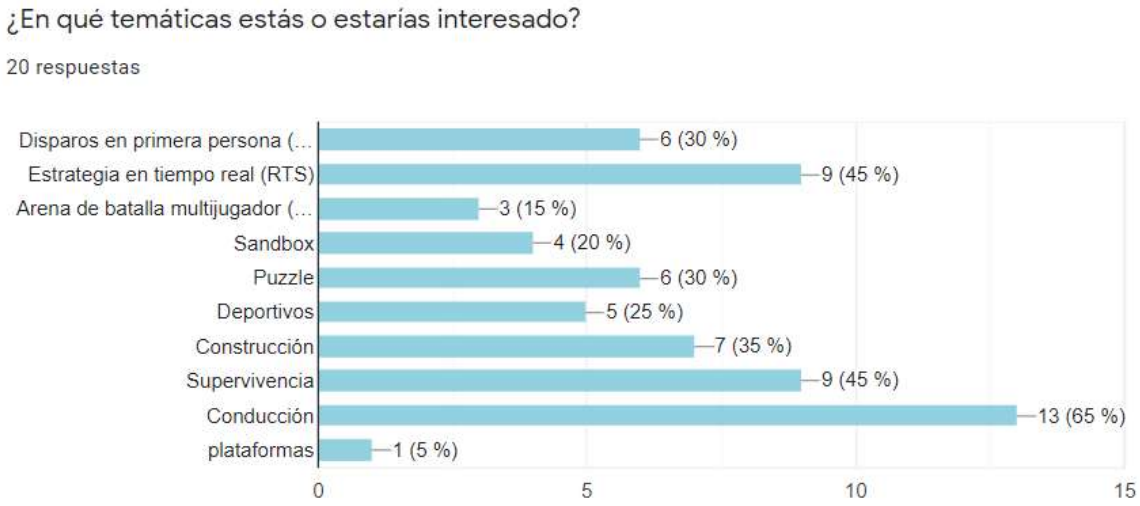
5.1.3. ¿Juegas frecuentemente a videojuegos?



Aquí se pretende saber cómo de familiarizado está el usuario con los videojuegos. Generalmente todos los géneros combinan un sistema de control parecido entre sí,

haciendo que los jugadores que comienzan un título por primera vez puedan hacerse una idea del control de una forma rápida, ágil y sin dificultades.

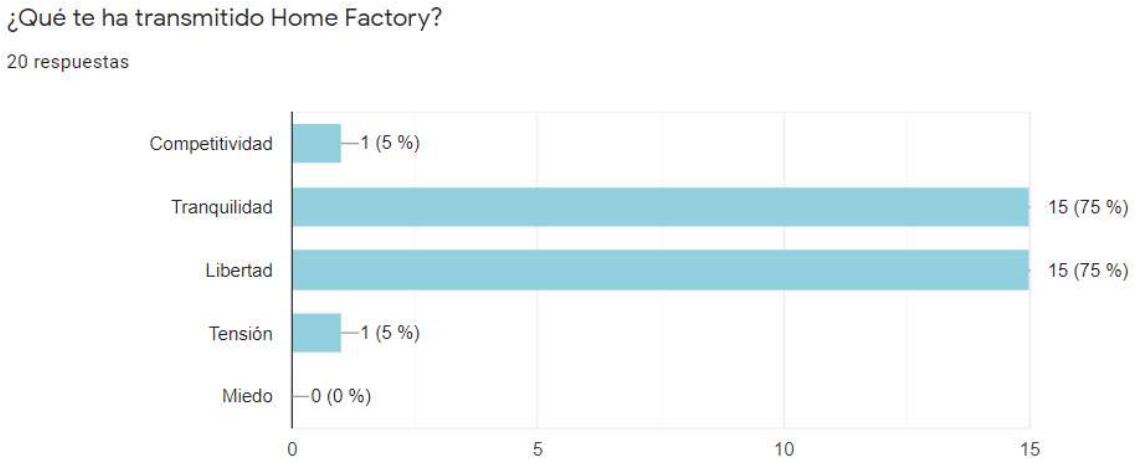
5.1.4. ¿En qué temática estás o estarías interesado?



Las temáticas en las que los jugadores estén interesados también suponen un dato importante y ayudan a entender cómo se van a desenvolver en Home Factory que, como ya sabemos, dispone de elementos de juegos de Disparos en primera persona (cámara, control, apuntado), Sandbox (libertad sin objetivo concreto), Construcción (construcción de estructuras), Supervivencia (controlar la componente económica para poder continuar jugando).

Se puede comprobar como casi la mitad usuarios están interesados en juegos de primera persona y supervivencia, y no tan experimentados en juegos sandbox o de construcción.

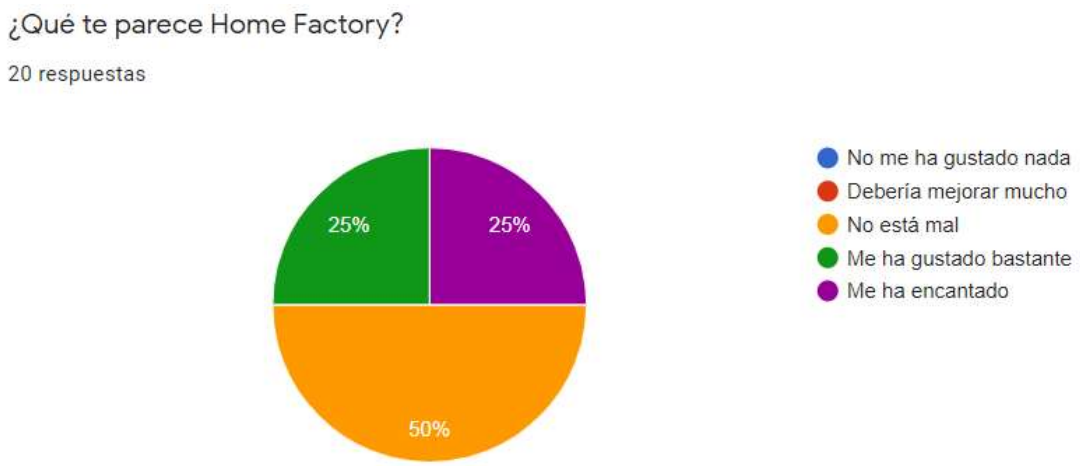
5.1.5. ¿Qué te ha transmitido Home Factory?



Uno de los principales objetivos de Home Factory era hacer un título que fuese capaz de transmitir tranquilidad y libertad al jugador, sin que éste tenga que estar preocupado de un tiempo, unos objetivos constantes, etc.

Según se puede comprobar en los resultados, la sensación de tranquilidad y libertad ha sido lo que más predomina por lo que se considera un objetivo conseguido.

5.1.6. ¿Qué te parece Home Factory?

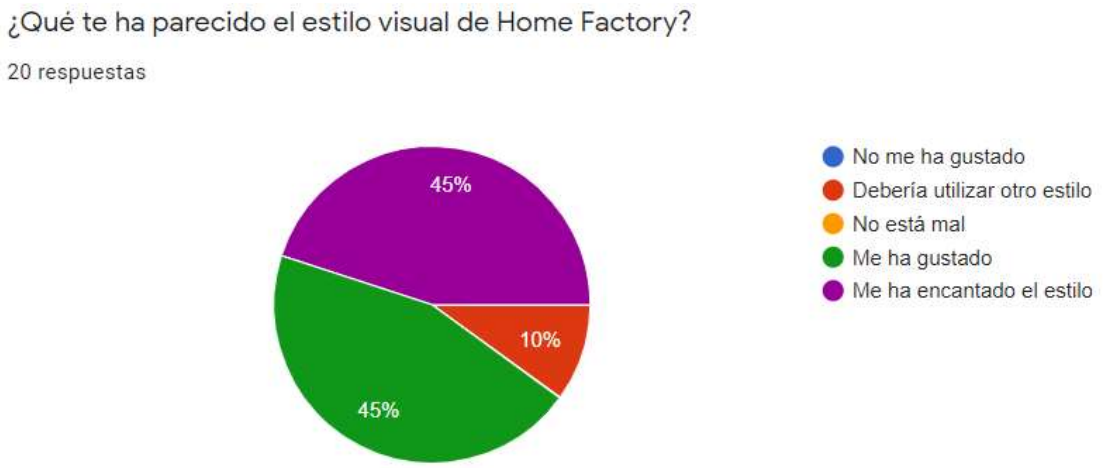


Sobre la opinión general del producto, se puede obtener bastante información. En primer lugar, todas las respuestas corresponden a un baremo neutral-positivo.

Por otro lado, y al tratarse de un prototipo, se entiende que el predominio de los resultados sea neutral de manera general aunque también hay personas que han disfrutado mucho de la idea y la han recibido de manera muy positiva.

En futuras versiones se podrá mejorar exponencialmente estos resultados obtenidos mediante las mejoras de fallos/bugs/rendimiento y aportes de la comunidad.

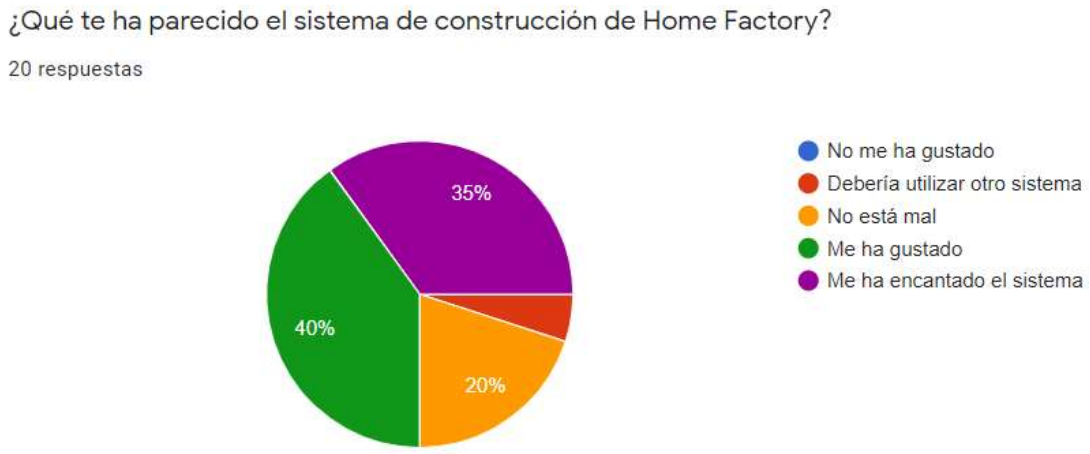
5.1.7. ¿Qué te ha parecido el estilo visual de Home Factory?



Como se puede comprobar, el estilo visual Cel Shading que utiliza Home Factory ya visto en otros títulos como Zelda: Breath of the Wild [79] o XIII [80] no es un estilo especialmente común que predomine en la industria, por lo tanto, se quería comprobar la aceptación de los usuarios.

El estilo visual de Home Factory ha sido muy bien recibido por la mayoría de los usuarios.

5.1.8. ¿Qué te ha parecido el sistema de construcción de Home Factory?

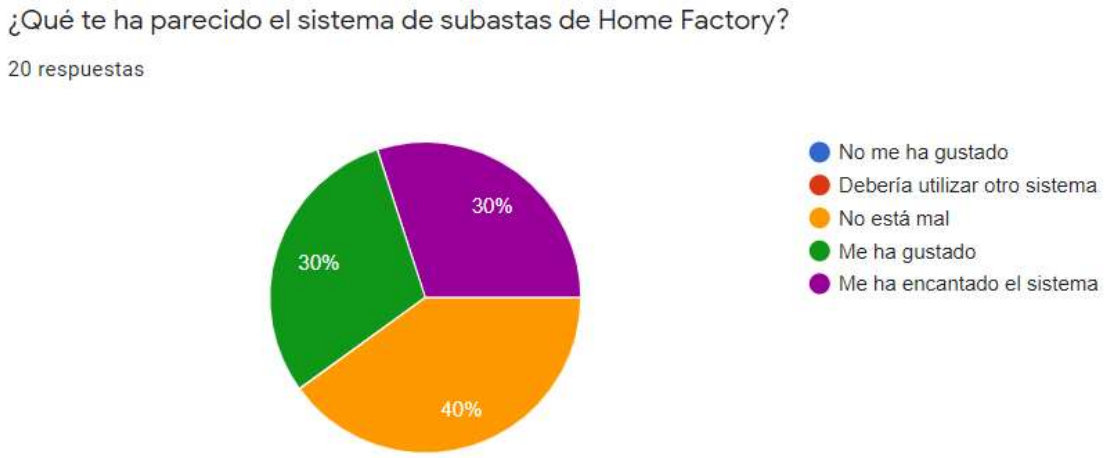


Otro de los apartados principales que se deseaba evaluar con los usuarios era el sistema de construcción.

Como ya se ha hablado en otros apartados, se trata de un sistema de construcción guiado mediante colores para indicar la viabilidad de la colocación de las piezas (verde, amarillo, rojo) y auto ajuste de piezas.

Según se puede comprobar en los resultados, ha sido un sistema mayoritariamente bien recibido. Se entiende que hace referencia en su mayoría al sistema de autoajuste y el hacer que todas las piezas encajen perfectamente entre sí convirtiéndose en uno de los puntos que más ha podido gustar a los jugadores sobre el sistema de construcción.

5.1.9. ¿Qué te ha parecido el sistema de subastas de Home Factory?



El sistema de subastas también es uno de los puntos importantes del juego, por lo que se consideró necesario integrarlo en el formulario.

Si bien es cierto que se trata de un prototipo para transmitir a los jugadores la idea de tener un sistema de subastas en juego, aún tiene mucho que mejorar y es muy posible haya sido percibido por los jugadores según los resultados obtenidos.

Sin embargo, se considera que la idea principal de un sistema de subastas ha sido bien acogida por los usuarios encontrados al encontrar un 60% de aceptación entre los encuestados, por lo que aún pendiente de mejora, se considera como otro objetivo conseguido.

5.1.10. ¿Qué no te ha gustado del juego?

La IU
Se centra bastante en la construcción
Si
Requerimientos previos para la instalación complicados de conseguir
Necesita mucho rendimiento para poder iniciar el videojuego, y que necesita más optimización.

no hay mucho contenido
Que en la vista aparezcan unos brazos.
Terreno con demasiados desniveles que dificultan bastante la construcción.
Optimización, lag.
Es muy pesado para mi equipo
Estética
Que es muy limitado en cuando a las construcciones que se pueden hacer.
La resolución de pantalla
no es posible construir en varios lugares diferentes
Poca variedad de elementos de construcción
nada
El césped dentro de la casa
Que hay algunas combinaciones que no se pueden realizar
Nada
Nada
Que no hay personajes

Sobre los resultados que no han gustado obtenidos se puede destacar la limitación de las construcciones y/o piezas disponibles y la optimización del título.

La limitación de las piezas se debe a que es un prototipo, aunque se tendrán en cuenta dichos comentarios de mejora para incorporar muchas más piezas y contendido en versiones posteriores.

Sobre la optimización, se va a dedicar un apartado completo de optimización del título en este capítulo bajo el título "**5.2 Pruebas de rendimiento**"

5.1.11. ¿Qué te ha gustado del juego?

Resulta muy relajante. La dinámica de construcción entretiene bastante.
Me ha gustado la tranquilidad que transmite, el estar en un bosque, escuchando a los pájaros mientras estás haciendo tus cosas
Empezar con Dinero para poder hacer cosas
La ejecución en sí misma

La sensación de tranquilidad al caminar etc., todo se veía muy real. El sistema de construcción es bueno por las animaciones etc.
se ve muy bien estéticamente
Los gráficos de la naturaleza y la música de ambiente.
La cantidad de elementos para construir es bastante alta.
Gráficos, movimientos, jugabilidad.
La idea de las subastas
jugabilidad
Tiene buenos gráficos y bastante potencial.
Las manitas jajja
fácil de manejar, se obtiene rápidamente una buena construcción
El entorno
todo
El paisaje
imaginarme en la casa que he construido en el lago
todo
el sistema de construcción
la creatividad

En este apartado se han recibido muchas valoraciones positivas que hacen que Home Factory se sienta como un producto con mucho potencial de desarrollo y aceptación entre los usuarios.

Se puede destacar los gráficos, las sensaciones que transmite, el concepto de subastas y la facilidad de construcción.

Se entiende por gráficos el sistema visual del título. El uso del Cel Shading parece haber llamado mucho la atención de los usuarios y haber sido muy bien recibido por los usuarios.

Las sensaciones que transmite son un punto a destacar muy importante ya que se trata de uno de los pilares del título y nos encanta ver cómo se ha transmitido la intención en su totalidad y la recepción que ha tenido por parte de los jugadores.

El sistema de subastas, como se ha indicado anteriormente, se trata sólo de una idea conceptual que se ha implementado en el título como elemento diferenciador, y además comprobar la recepción por parte de los usuarios de este sistema. Al parecer ha sido bien recibido como para seguir con su desarrollo de una manera más detallada.

Que el sistema de construcción sea sencillo para los usuarios es algo importante y que va ligado con el apartado sensaciones que queríamos transmitir al jugador desde el inicio. Si se tratase de un sistema complejo, difícil de entender y de interactuar con él se opondría con nuestro objetivo primario de transmitir tranquilidad y libertad al jugador. En este caso parece que el sistema de construcción y su auto-ajuste ha tenido una buena recepción por parte de los jugadores.

5.1.12. ¿Qué te gustaría ver en versiones posteriores?

Más repertorio en las opciones de construcción. También sería interesante tener distintas opciones en la iluminación exterior (noche, atardecer, etc.)
me encantaría ver un sistema de combate donde poder recoger dinero o elementos para poder utilizarlos en la construcción
Un pequeño mensaje al inicio de cómo conseguir recursos una vez que se te acabe el dinero inicial.
Evolución del juego en general
Nuevos personajes o mobs.
Más construcciones
Una mayor variedad de elementos de construcción.
Un mejor sistema de apuntado para la colocación de objetos que estén muy cerca unos de otros.
Mejor optimización, versión móvil.
Multi usuarios
Si
Muchas más estructuras para construir.
Mas opciones de construcción
Tener la posibilidad de construir más libremente
Más variedad

Muebles para decorar el interior de la casa
Objetos para decorar
Muebles
Muñecos y cosas de Disney
Decoración interior
Que hubiera personajes

De entre todas las solicitudes de mejora, se puede destacar: mensaje al inicio de cómo conseguir recursos / dinero, mejor optimización y más elementos de construcción.

El mensaje al inicio de cómo conseguir recursos/dinero se trata de una idea genial para aquellos jugadores que no estén tan familiarizados con la jugabilidad de estos títulos o que no hayan sido capaces, por una u otra razón de llegar al punto de subasta para conseguir recursos/dinero. Se tratará posteriormente con las ideas de implementación de tutoriales y mejora de interfaz en el "**Capítulo 6 – Conclusiones**"

La optimización del título parece ser que ha resultado uno de los puntos clave a mejorar del título y que se debía tratar lo antes posible antes de adentrarnos en otro tipo de mejoras en la jugabilidad. Se ha dedicado un apartado completo a la parte de optimización en este mismo capítulo bajo el título "**5.2 Pruebas de rendimiento**".

Más elementos de construcción: Otro de los puntos clave a tener en cuenta como mejora y evolución de Home Factory. Para el prototipo se desarrollaron las piezas que consideraron principales y más importantes para su publicación, aunque se entiende que aún hay muchos más elementos que pueden ser incorporados. Se desarrollará este punto en el "**Capítulo 6 – Conclusiones**"

En el "**Capítulo 6 – Conclusiones**" no sólo se tratarán las conclusiones que ha supuesto la realización de Home Factory, sino además de todos aquellos puntos de mejora e ideas de evolución para Home Factory.

5.2. Pruebas de rendimiento

Tanto el desarrollo como el testeo de Home Factory se ha realizado en un equipo con las siguientes características técnicas:

HP OMEN 15-dh0018ns:

- Intel Core i7 – 9750H
- 16GB de RAM
- 1 TB SSD M.2
- nVidia GeForce RTX 2060-6GB
- Pantalla 15.6 pulgadas FullHD, 240Hz

Aunque se trata de un portátil capaz de ofrecer grandes capacidades de rendimiento, se han experimentado algunos problemas en el nivel principal del videojuego.

El título ha sido perfectamente jugable en el PC de desarrollo indicado, sin embargo, ofrecía un framerate mínimo de ~30fps y máximo de ~46fps cuando se apuntaba con el jugador al horizonte.

Ilustración 84 - Framerate mínimo sin optimización

Cuando se apuntaba a zonas donde no se tenían que cargar tantos elementos en pantalla (apuntar al suelo, por ejemplo), el título llegaba a ofrecer hasta 85 fps.

Ilustración 85 - Framerate máximo sin optimización

Como se puede comprobar el título es perfectamente jugable en un equipo de características similares o superiores a las indicadas, sin embargo, se convertía en una limitación de optimización para equipos de rendimiento inferior.

Según los comentarios recibidos en las encuestas, se pretendía mejorar el rendimiento de Home Factory en todos los equipos independientemente de la gama a la que pertenezcan.

Al tratarse de un escenario de esas dimensiones, y con toda la carga de foliage que tiene el mapeado, la propiedad que más nos interesaba ajustar era "**Cull Distance**". Esta propiedad permite indicar la distancia a partir de la cual los objetos del mapa, el foliage en este caso, no serán renderizados. De esta forma, se liberará la carga gráfica de Home Factory ya que sólo se crearán los elementos necesarios según cercanía y posición del jugador.

Se debe tener especial atención con esta propiedad ya que asignar valores demasiado bajos, incrementará radicalmente el rendimiento, sin embargo, se comenzarán a sufrir efectos de popping, es decir, se empezará a ver de forma muy acentuada cómo se van creando estos objetos creando una experiencia muy artificial y negativa para el jugador.

Ilustración 86 - Ejemplo Cull max distance 100 para todo foliage

La idea de ajustar el Cull Distance es buscar el punto intermedio donde los elementos que están fuera del campo de visión del jugador no sean generados para ahorrar carga a la tarjeta gráfica, pero sí aquellos cuya distancia sea visible por el jugador para evitar transmitir esa sensación de que el mundo se va creando a su paso. Tras ajustar el parámetro "**Max cull distance**" de todos los elementos que componen el escenario, se ha mejorado considerablemente el rendimiento pasando

de 30 fps de mínima a unos 62 fps de mínima, es decir, un 100% de incremento de mejora sobre los resultados recibidos inicialmente.

Ilustración 87 - Framerate mínimo tras optimización

Respecto a la tasa máxima de fps, se ha conseguido aumentar desde 85fps hasta los 135fps, es decir, un 58,82% para su tasa máxima.

Ilustración 88 - Framerate máximo tras optimización

Estas mejoras de HomeFactory han conseguido que el juego aumente considerablemente su compatibilidad con aquellos equipos de inferiores prestaciones que podrían presentar un rendimiento muy por debajo del esperado y experimentado durante el desarrollo o incluso incapacidad para poder disfrutar del título.

Capítulo 6 – Conclusiones

En este apartado se hablará acerca de las conclusiones obtenidas tras el desarrollo al completo de Home Factory, lecciones aprendidas e ideas para próximos desarrollos y ampliaciones del título.

6.1. Logros alcanzados

Tal y como se puede comprobar en este apartado, los objetivos generales y específicos descritos en el punto **2.1 Objetivos**, han sido cumplidos satisfactoriamente: Se ha desarrollado un prototipo de un videojuego de construcción de estructuras con mecánicas de subasta para la gestión de las mismas y que se diferencie de los videojuegos ya existentes, se desarrolla una funcionalidad de construcción y estructuras, se proporciona al jugador la capacidad de gestionar su dinero inicial para crear y editar las estructuras, se incorpora una interfaz de gestión atractiva e intuitiva para el jugador y se asigna a la estructura del jugador una puntuación que determine valores aproximados en la fase de subasta.

Además de los requisitos cumplidos, se ha realizado un estudio de las herramientas para el desarrollo de Home Factory (Unity vs Unreal Enigne 4), se han empleado aquellas herramientas de creación de geometría más eficientes y adaptadas a los requerimientos del título (haciendo uso de la herramienta de geometría que incorpora Unreal Engine 4), se ha seguido un riguroso proceso de implementación mediante iteraciones que ha conseguido que Home Factory fuese creciendo de una forma segura, comprobando que toda nueva implementación no interfiriese con las ya existentes y puedan convivir sin incompatibilidades; se ha conseguido transmitir al jugador la sensación de tranquilidad y libertad según hemos podido comprobar en los resultados y todas las ideas implementadas en el título (el sistema de construcción, el estilo visual y el sistema de subastas) han sido bien acogidas por los usuarios.

6.2. Lecciones aprendidas

Durante la realización de Home Factory, ha habido determinados factores que han supuesto una lección como desarrollador y que pueden resultar como puntos a tener en cuenta para otros desarrolladores.

En mi situación particular, una vez terminada la carrera en 2018 se postpuso la realización del TFG o Trabajo de Fin de Grado, lo que supuso varios factores a lo largo del tiempo. Primero, pérdida de entusiasmo y ganas para su realización, ya que, al no necesitar formalizar la finalización del mismo para continuar con mi día a día en el terreno profesional, dar este paso nunca resultó ser un requisito, por lo que se aplazaba constantemente dando lugar a que cada vez más mis conocimientos acerca de la materia desde el punto de vista de desarrollador sean cada vez más lejanos y limitados.

El confinamiento, el apoyo de las personas cercanas, un replanteo de futuro, y más tiempo libre del que me gustaría tener fueron elementos clave para ayudar a terminar este punto pendiente en mi vida. Además, aparece la oportunidad de realizar un curso de infoarquitectura en Unreal Engine algo que, aun no siendo directamente relacionado con videojuegos y con el proyecto que pudiese tener en la cabeza, sirvió para recordar el uso a nivel medio de la herramienta para, a partir de este punto, empezar a despegar la idea de lo que sería Home Factory.

Llegar a este punto de distanciamiento con la materia supone dos elementos clave en el desarrollo del título, por un lado la obligación del autoaprendizaje continuo mientras se avanza en el desarrollo del título, descubrir cuál es la mejor base para empezar y cuáles son los elementos adecuados para continuar con ese desarrollo de forma modular; y por otro lado, el desconocimiento al no haber trabajado nunca en un proyecto de semejante envergadura incapacita el correcto dimensionamiento del proyecto a realizar, abarcando por ello en ocasiones un proyecto de mayor ambición del que en realidad debería ser.

En el caso particular de Home Factory y su principal pilar, la construcción, el no tener claro cómo sería la metódica empleada para la construcción también hace que ello suponga un retraso en el desarrollo. Preguntas tales como ¿cuál sería el método correcto para desarrollar Snap-In de la manera más eficiente y funcional posible?

¿Se debería implementar un sistema de auto-ajuste? ¿Se debería dar más libertad al usuario? ¿Cuál es el tamaño óptimo para las piezas? ¿Encajan todas a la perfección al realizar una construcción de tipo X o tipo Y? Al final se encontraron respuesta a estas y otras muchas preguntas que fueron surgiendo durante el desarrollo de Home Factory aunque quizá haber tenido todo mucho más desarrollado en papel hubiese hecho de algunas partes un desarrollo mucho más eficiente si todo hubiese estado ya precalculado, sobre todo los tamaños de las piezas respecto al personaje y cómo encajan entre sí.

El resultado final de Home Factory, aun tratándose de un prototipo, resulta tremendamente satisfactorio, aunque se considera muy conveniente en este apartado compartir aquellas experiencias que puedan ser puntos referentes a tener en cuenta por aquellos alumnos en situaciones similares al haber terminado toda la parte teórica del grado, pudiendo resumir las lecciones aprendidas en 5 puntos clave:

- **Encuentra tu motivo para finalizarlo:** Cada uno tiene el suyo, sólo tienes que encontrar el tuyo. Sea o no sea un requisito para tu día a día profesional, termina todo aquello que empieces
- **No aplaces su realización más de lo necesario:** Como con cualquier cosa en la vida, siempre seremos capaces de encontrar una excusa para no empezar hoy.
- **Aprovecha de los conocimientos que ya tienes:** Ser autodidacta es muy bueno y aunque se aprendan muchos conceptos durante la carrera, será necesario en muchas ocasiones, pero será mucho más fácil cuando estos conceptos hayan formado parte de tu día a día recientemente que años después.
- **Dimensiona tu proyecto siendo realista:** Si el desarrollo se va a realizar por una única persona, no dimensiones más allá de lo que consideres que puedas realizar. Trata de fijar una fecha aproximada y piensa hasta dónde eres capaz de llegar en ese tiempo.
- **Comienza diseñando en papel:** Intenta tener claro todos los puntos principales de tu proyecto en papel, especialmente si jugar con las medidas va a ser uno de los puntos fuertes en tu proyecto, intenta tener todo claro

antes de ponerte manos a la obra y tener que replantear las medidas geométricas varias ocasiones.

Dicho esto, considero la realización del TFG como una auténtica experiencia positiva que me ha enseñado no sólo estos puntos de lecciones aprendidas, sino hasta dónde soy capaz de llegar empezando en un área prácticamente desde cero y que, aun habiendo insistido tras la carrera que el Diseño y Desarrollo de Videojuegos "no es lo mío", una vez habiéndome visto frente al ordenador tantas horas, días y noches, si se me presentase la oportunidad de dedicarme a ello profesionalmente, la aceptaría sin duda alguna.

6.3. Ideas de futuro para Home Factory

Al tratarse de un prototipo, hay numerosas ideas que han ido surgiendo a lo largo del desarrollo que han decidido aplazarse para su implementación en futuras versiones del producto, así como aquellas ideas recibidas por el feedback de los usuarios que han tenido ocasión de testearlo.

6.3.1. Mejoras en la optimización

Tal vez uno de los elementos más importantes como idea de futuro. Aunque se ha dedicado un apartado a la optimización del título, aún habría que comprobar, según requisitos, qué tal se desenvuelve Home Factory en equipos de características limitadas para así poder optimizar aún más y ampliar su público.

Una de las mejoras que se desearía implementar es la gestión de los actores en el escenario. Crear y eliminar constantemente elementos en el escenario, es una acción que supone un coste computacional, sin embargo, aquellos elementos que sean eliminados (con la función de reembolso) podrían pasar a ser no visibles y cuando el jugador vuelva a utilizar una pieza de ese tipo, volver a hacerla visible en las coordenadas donde el jugador está colocando la pieza. De esta forma podríamos aliviar el coste computacional.

6.3.2. Más elementos de construcción

Otra característica que se desea implementar en Home Factory y que ya se estuvo intentando desarrollar anteriormente (revisar apartado de dimensionar el proyecto siendo realista) es el de incorporar más elementos de construcción y un apartado completo de decoración.

Armarios, camas, lámparas, bañeras, duchas, neveras, hornos, encimeras, etc. Cualquier tipo elemento para poder decorar nuestra casa al completo y a nuestro gusto, con las correspondientes texturas para poder realizar cambios de color.

Además, se desearía implementar que toda fuente artificial de luz pueda ser interactiva para el usuario.

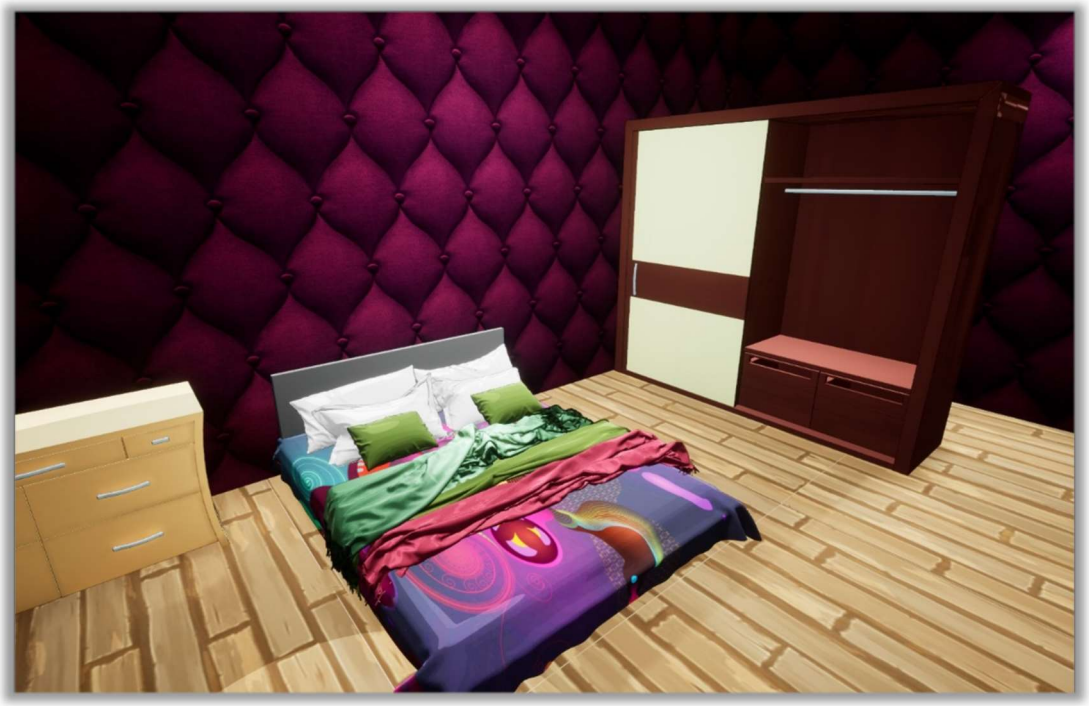


Ilustración 89 - Prototipo de Home Factory con elementos decorativos

6.3.3. Ciclos día/noche

Al igual que la implementación de más elementos de construcción, se desearía implementar ciclos de día/noche para que el jugador pueda ver su construcción y cómo la luz natural interactúa con la estructura en los diferentes ciclos del día.

6.3.4. Más personajes

Otra de las sugerencias que Home Factory ha recibido en las encuestas ha sido la implementación de más personajes. Podría ser interesante tener NPCs distribuidos por el mapa con los que se puedan interactuar para el desbloqueo de piezas, elementos de decoración y/o texturas.

6.3.5. Múltiples huecos de guardado

En esta versión sólo se dispone de un hueco de guardado activo a la vez, sin embargo, se desearía desarrollar un sistema de múltiples huecos de guardado para que el jugador pueda decidir sobre cuál de ellos desea interactuar y cargar partida.

6.3.6. Mejora del sistema de subastas

El sistema de subastas aun siendo un elemento muy bien recibido en las encuestas, tiene mucho desarrollo pendiente como ya se ha mencionado con anterioridad. Desde tener más agentes, a mejorar el algoritmo económico o incluso meter una vertiente multijugador opcional donde los jugadores puedan interactuar sólo con las subastas activas de otros jugadores.

Es decir, una vez que el jugador termina su creación la publica en un servidor con un límite de tiempo activa, y los jugadores que estén interesados pueden previsualizar la estructura publicada y hacer una puja por ella.

Además, se desearía implementar que las texturas y elementos de decoración interfieran en el valor de la casa.

6.3.7. Edición del terreno

Quizá el punto más complejo a implementar junto con la mejora del sistema de subastas multijugador. Aun así, se desearía implementar un "modo terreno" donde el jugador pueda interactuar con el terreno: aumentar o disminuir altura, encontrar río, eliminar o crear árboles, para poder acondicionar aún más su estructura.

Bibliografía

[1] «Namco Bandai,» [En línea]. Available: https://es.bandainamcoent.eu/.

[2] G. MX, «Pac-Man cumple 39 años!,» [En línea]. Available: https://www.golsystems.mx/2019/05/21/pac-man-cumple-39-anos/.

[3] microsiervos, «Space Invaders: La historia de su creación y de cómo se convirtió en un icono del diseño,» [En línea]. Available: https://www.microsiervos.com/archivo/juegos-y-diversion/space-invader-historia-creacion-icono-diseno.html.

[4] Archive.org, «K.C. Munchkin (1982) (US),» [En línea]. Available: https://archive.org/details/K.C._Munchkin_1982_Philips_US.

[5] MobyGames, «Brøderbund Software, Inc.,» [En línea]. Available: https://www.mobygames.com/company/brderbund-software-inc.

[6] MobyGames, «Lode Runner,» [En línea]. Available: https://www.mobygames.com/game/lode-runner.

[7] Nintendo, «Excitebike,» [En línea]. Available: https://www.nintendo.es/Juegos/NES/Excitebike-751574.html.

[8] VidaExtra, «Famicom Cumple 30 años,» [En línea]. Available: https://www.vidaextra.com/otras-plataformas/famicom-cumple-30-anos-cinco-curiosidades-que-demuestran-que-no-hemos-cambiado.

[9] VidaExtra, «'Raid on Bungeling Bay', el primer juego de Will Wright,» [En línea]. Available: https://www.vidaextra.com/accion/raid-on-bungeling-bay-el-primer-juego-de-will-wright.

[10] InsertCoin, «SimCity (1989),» [En línea]. Available: https://www.insertcoinclasicos.com/2006/08/25/sim-city-1989/.

[11] NoSoloBits, «Gremlin Interactive,» [En línea]. Available: https://www.nosolobits.com/es/compania/256/gremlin-interactive.

[12] NoSoloBits, «Virgin Interactive,» [En línea]. Available: https://www.nosolobits.com/es/compania/56/virgin-interactive.

[13] GameFAQs, «Greg Norman's Golf Power,» [En línea]. Available: https://gamefaqs.gamespot.com/nes/587315-greg-normans-golf-power.

[14] Steam, «Wolfenstein 3D,» [En línea]. Available: https://store.steampowered.com/app/2270/Wolfenstein_3D/.

[15] Fandom, «MapEdit,» [En línea]. Available: https://wl6.fandom.com/wiki/MapEdit.

[16] Bethestda, «Doom (1993),» [En línea]. Available: https://bethesda.net/es/store/product/DO1GNGPCBG01.

[17] Steam, «QUAKE,» [En línea]. Available: https://store.steampowered.com/app/2310/QUAKE/.

[18] Fandom, «Doom Editing Utilities,» [En línea]. Available: https://doom.fandom.com/wiki/Doom_Editing_Utilities.

[19] Fandom, «QuakeEd,» [En línea]. Available: https://quake.fandom.com/wiki/QuakeEd.

[20] Steam, «RollerCoaster Tycoon®: Deluxe,» [En línea]. Available: https://store.steampowered.com/app/285310/RollerCoaster_Tycoon_Deluxe/.

[21] Comiqueros, «Retro Review: Tony Hawk's Pro Skater 2 – El apogeo del Skate,» [En línea]. Available: https://comiqueros.cl/retro-review-tony-hawks-pro-skater-2-el-apogeo-del-skate/.

[22] Meristation, «Neverwinter Nights,» [En línea]. Available: https://as.com/meristation/juegos/neverwinter_nights/1/.

[23] Bioware, «Bioware,» [En línea]. Available: https://www.bioware.com/.

[24] IGN, «La Historia de Baldur's Gate,» [En línea]. Available: https://es.ign.com/baldurs-gate-enhanced-edition-pc/61965/feature/la-historia-de-baldurs-gate.

[25] D. &. Dragons, «Dungeons & Dragons,» [En línea]. Available: https://dnd.wizards.com/.

[26] N. Wiki, «Aurora Toolset,» [En línea]. Available: https://nwn.wiki/display/NWN1/Aurora+Toolset.

[27] C. P. Red, «The Witcher,» [En línea]. Available: https://thewitcher.com/en/witcher1.

[28] W. o. warcraft, «Warcraft III: Reign of Chaos,» [En línea]. Available: https://worldofwarcraft.com/es-es/story/timeline/chapter-04.

[29] B. Entertainment, «Blizzard Entertainment,» [En línea]. Available: https://www.blizzard.com/es-es/.

[30] Wikipedia, «Defense of the Ancients,» [En línea]. Available: https://es.wikipedia.org/wiki/Defense_of_the_Ancients.

[31] W. o. Warcraft, «Warcraft III: The Frozen Throne,» [En línea]. Available: https://worldofwarcraft.com/es-es/story/timeline/chapter-5.

[32] R. Games, «League of Legends,» [En línea]. Available: https://na.leagueoflegends.com/es-es/.

[33] Valve, «DOTA 2,» [En línea]. Available: https://www.dota2.com/home.

[34] Steam, «Garry's Mod,» [En línea]. Available: https://store.steampowered.com/app/4000/Garrys_Mod/.

[35] Steam, «Half-Life 2,» [En línea]. Available: https://store.steampowered.com/app/220/HalfLife_2/.

[36] Bungie, «Bungie,» [En línea]. Available: https://www.bungie.net/ES.

[37] 3djuegos.com, «Análisis de Halo 3,» [En línea]. Available: https://www.3djuegos.com/juegos/analisis/1273/0/halo-3/.

[38] Microsoft, «Minecraft,» [En línea]. Available: https://www.minecraft.net/es-es/.

[39] Computteren, «Minecraft: Redstone Tutorials - All 16 Logic Gates,» [En línea]. Available: https://www.youtube.com/watch?v=ambUBcVMyn0.

[40] CaptainSparklez, «Minecraft: Working Cell Phone w/ Web Browser and Video Calling,» [En línea]. Available: https://www.youtube.com/watch?v=IdlZRhKmWJY.

[41] Terraria, «Terraria,» [En línea]. Available: https://terraria.org/.

[42] E. Games, «Epic Games,» [En línea]. Available: https://www.epicgames.com/store/es-ES/.

[43] E. Games, «Fortnite,» [En línea]. Available: https://www.epicgames.com/fortnite/es-ES/home.

[44] PUBG, «PUBG,» [En línea]. Available: https://www.pubg.com/es/.

[45] Steam, «Prison Architect,» [En línea]. Available: https://store.steampowered.com/app/233450/Prison_Architect/.

[46] Facepunch, «Rust,» [En línea]. Available: https://rust.facepunch.com/.

[47] 3djuegos.com, «Análisis de Fallout 4. Un Apocalípsis inabarcable,» [En línea]. Available: https://www.3djuegos.com/juegos/analisis/20923/0/fallout-4/.

[48] H. Games, «No Man's Sky NEXT,» [En línea]. Available: https://www.nomanssky.com/next-update/.

[49] Steam, «ARK: Survival Evolved,» [En línea]. Available: https://store.steampowered.com/app/346110/ARK_Survival_Evolved/.

[50] P. Interactive, «Cities: Skylines,» [En línea]. Available: https://www.citiesskylines.com/.

[51] Natsume, «Harvest Moon,» [En línea]. Available: http://www.natsume.com/.

[52] E. Barone, «Stardew Valley,» [En línea]. Available: https://www.stardewvalley.net/.

[53] Nintendo, «Animal Crossing: New Horizons,» [En línea]. Available: https://www.animal-crossing.com/new-horizons/es/.

[54] Nintendo, «Super Mario Maker 2,» [En línea]. Available: https://supermariomaker.nintendo.com/es/.

[55] Blizzard, «Diablo III,» [En línea]. Available: https://eu.diablo3.com/es-es/.

[56] VentureBeat, «Meet the gamers who earned big in the now-closed Diablo III real-money auction house,» [En línea]. Available: https://venturebeat.com/2014/03/19/meet-the-gamers-who-earned-big-in-the-now-closed-diablo-iii-real-money-auction-house/.

[57] Blizzard, «World of Warcraft,» [En línea]. Available: https://worldofwarcraft.com/es-es/.

[58] M. Studios, «Forza Horizon 4,» [En línea]. Available: https://www.xbox.com/es-ES/games/forza-horizon-4.

[59] E. Sports, «FIFA 20: Ultimate Team,» [En línea]. Available: https://www.ea.com/es-es/games/fifa/fifa-20/ultimate-team/features.

[60] 2K, «NBA 2K21,» [En línea]. Available: https://nba.2k.com/es-ES/.

[61] U. Technologies, «Unity,» [En línea]. Available: https://unity.com/.

[62] E. Games, «Unreal Engine,» [En línea]. Available: https://www.unrealengine.com/en-US/.

[63] BaboonLab, «Blueprints en Unreal Engine 4, funciones y tipos,» [En línea]. Available: https://baboonlab.odoo.com/blog/noticias-de-marketing-inmobiliario-y-tecnologia-1/post/blueprints-en-unreal-engine-4-funciones-tipos-y-otras-caracteristicas-21.

[64] U. E. Docs, «UMG UI Designer,» [En línea]. Available: https://docs.unrealengine.com/en-US/InteractiveExperiences/UMG/index.html.

[65] Adobe, «Photoshop,» [En línea]. Available: https://www.adobe.com/es/products/photoshop.html.

[66] E. Games, «Unreal Engine Marketplace,» [En línea]. Available: https://www.unrealengine.com/marketplace/en-US/store.

[67] U. Engine, «UE4 Answer Hub,» [En línea]. Available: https://answers.unrealengine.com/index.html.

[68] Google, «YouTube,» [En línea]. Available: https://www.youtube.com/?gl=ES.

[69] Evermotion, «Archviz Vol. 1,» [En línea]. Available: https://evermotion.org/shop/show_product/archinteriors-for-ue-vol-1/12025.

[70] Microsoft, «Excel,» [En línea]. Available: https://www.microsoft.com/es-es/microsoft-365/excel.

[71] TomkaGS, «Stylized Forest,» [En línea]. Available: https://www.unrealengine.com/marketplace/en-US/product/stylized-forest-03.

[72] LowlyPoly, «Stylized Texture Pack - VOL.05 Hand Painted Textures,» [En línea]. Available: https://www.unrealengine.com/marketplace/en-US/product/stylized-texture-pack-vol-05-hand-painted-textures.

[73] S. Studios, «Cartoon Cel Shader,» [En línea]. Available: https://www.unrealengine.com/marketplace/en-US/product/cartoon-cel-shader.

[74] G.-R. Characters, «Female Hands,» [En línea]. Available: https://www.unrealengine.com/marketplace/en-US/product/female-hands/.

[75] bauyrzhanmustafin, «Stylized VFX Pack,» [En línea]. Available: https://www.unrealengine.com/marketplace/en-US/product/stylized-vfx-pack.

[76] A. P. Sheets, «Butter-Fly (Full ver.) - Digimon Adventure OP [Piano],» [En línea]. Available: https://www.youtube.com/watch?v=T_D_pp283nw.

[77] L. Stirling, «Zelda Medley - Lindsey Stirling,» [En línea]. Available: https://www.youtube.com/watch?v=T_D_pp283nw.

[78] H. N. &. Meditation, «GOOD MORNING SPRING NATURE THERAPY ASMR Ambience Meditation 4k Flowery Meadow Birds Sounds for sleep,» [En línea]. Available: https://www.youtube.com/watch?v=UZ9uyQI3pF0.

[79] «Zelda: Breath of the Wild,» Nintendo, [En línea]. Available: https://www.nintendo.es/Juegos/Nintendo-Switch/The-Legend-of-Zelda-Breath-of-the-Wild-1173609.html.

[80] «XIII Steam,» Microids, [En línea]. Available: https://store.steampowered.com/app/1154790/XIII/?l=spanish.