

TFG

DESAFIO EN LAS RUINAS PROHIBIDAS

DESARROLLO DE GRÁFICOS 3D PARA UN PROTOTIPO DE NIVEL DE VIDEOJUEGO DE AVENTURA/ACCIÓN EN UN CONTEXTO DE FANTASÍA MEDIEVAL.

Presentado por Vicente García García

Tutor: Francisco Martí Ferrer

Facultat de Belles Arts de Sant Carles

Grado en Bellas Artes

Curso 2019-2020



UNIVERSITAT
POLITÀCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA
FACULTAT DE BELLES ARTS DE SANT CARLES

Resumen

Este trabajo aborda fundamentalmente la elaboración de un escenario complejo para un prototipo de videojuego 3D de creación propia. No obstante, con objeto de obtener un prototipo jugable, he abordado parcialmente aspectos como mecánicas, programación, modelado de personajes, *rigging* y animación.

El modelado de los elementos del entorno y de los personajes se ha realizado con aplicaciones informáticas de modelado 3D y se han implementado en Unreal Engine IV, optimizando las técnicas de representación gráfica para que funcionasen correctamente en tiempo real.

El juego está ambientado en un entorno de fantasía, donde controlamos a nuestro protagonista, un ser de luz, a través de un mundo corrupto e infectado por seres malvados y oscuros que buscan la luz para absorberla. El protagonista busca adentrarse en las ruinas prohibidas, ya que estas han sido invadidas por hordas de monstruos que están absorbiendo la energía de este sitio sagrado y debilitando su núcleo de cristal, fuente de esta energía y objetivo final del juego.

Esta ambientación y la propia idea del videojuego atienden a mi necesidad por crear un escenario donde poder aplicar mecánicas como las batallas o el sistema de escalada que he creado para este proyecto.

El prototipo se centra en la exploración de un terreno poblado de diversos elementos. Para superar el nivel, el jugador deberá encontrar 50 monedas escondidas y llegar hasta el núcleo de cristal sin que los enemigos le derroten.

Palabras clave

Motor de juego, modelado 3d, Diseño de nivel, Mecánicas de juego, *rigging*, animación3d, programación, optimización.

Abstract

This project fundamentally deals with the elaboration of a complex scenario for a prototype of a 3D video game of its own creation. However, in order to obtain a playable prototype, I have partially addressed aspects such as mechanics, programming, character modeling, rigging and animation.

The modeling of the elements of the environment and the characters has been made with 3D modeling computer applications and have been implemented in Unreal Engine IV, optimizing the graphic representation techniques so that they work correctly in real time.

The game is set in a fantasy environment, where we control our protagonist, a being of light, through a corrupt world and infected by evil and dark beings who seek the light to absorb it. The protagonist seeks to enter the forbidden ruins, as these have been invaded by hordes of monsters that are absorbing the energy of this sacred site and weakening its crystal nucleus, source of this energy and final goal of the game.

This environment and the idea of the videogame address my need to create a scenario where I can apply mechanics such as the battles or the climbing system that I have created for this project.

The prototype focuses on the exploration of a terrain populated by various elements. To overcome the level, the player must find 50 coins hidden and reach the crystal core without enemies defeating him.

Keywords

Game engine, 3d modeling, level design, game mechanics, rigging, animation3d, programming, optimization.

Agradecimientos

A mi familia por toda la fuerza que me han dado a lo largo de esta etapa de mi vida.

A los profesores de BBAA, por guiarme en el camino correcto, enseñarme a ser mejor artista y persona.

A mi tutor Francisco Martí por su paciencia conmigo y por la gran ayuda que me ha aportado.

Índice

1. Objetivos y metodología	6
1.1. Objetivos	6
1.2 Metodología	7
2. Contexto y referentes	10
2.1 El género de acción/aventura	11
2.2 Referentes	13
3. Concepto y desarrollo del proyecto	14
3.1. Concept art / dibujos previos	14
3.2. Modelado del terreno	14
3.2.1. Vegetación y Rocas	14
3.2.2. Otros elementos del terreno	16
3.2.2 Físicas y colisiones	17
3.3 Interfaz y controles	18
3.4 Diseño de nivel y mecánicas	19
3.5. HUD	20
3.6 Personajes	22
3.6.1. Modelado	22
3.6.1.1 Personaje principal.	22
3.6.1.2. Enemigos	26
3.6.2 <i>Rigging</i> y <i>skinning</i> .	26
3.6.3. Animaciones	30
3.6.4. Programación y Jugabilidad	30
3.7 Retoques finales y futuras mejoras	33
4. Conclusiones	34
5. Referencias	35

1. Objetivos y metodología

1.1. Objetivos

Los objetivos de este trabajo son:

- Idear y desarrollar un prototipo para un nivel de videojuego 3d del género acción/aventura en tercera persona en el que poder conjugar la exploración de un entorno y las peleas en tiempo real, para enriquecer la experiencia en el juego y mantener el interés del jugador.

- Integrar los conocimientos adquiridos durante el Grado en diversos aspectos necesarios para la creación de un videojuego: *concept art*, modelado 3D, *rigging*, animación, programación y sonido.

- Profundizar en habilidades que me permitan desarrollar un perfil profesional adecuado tanto para el desarrollo de videojuegos independientes como para artista generalista en proyectos de mayor envergadura.

- Partiendo de mi experiencia previa con el motor de juego “Unity3D”, aprender a utilizar el motor de juego “Unreal Engine 4” mediante el desarrollo de un proyecto.

- Obtener un resultado apropiado para integrar en mi portfolio, con objeto de presentarlo en entornos académicos y profesionales.

Introducción

Siempre me ha interesado, como se plasma en proyectos anteriores en formato cómic¹, contar historias a partir de generar un mundo en el que el jugador se vea inmerso, de manera que se vaya adentrando en su historia y descubriendo los mensajes que esta esconde, sin necesidad de introducir textos explicativos.

En el caso de este trabajo, quiero trasladar ese modo de hacer a un formato narrativamente no lineal, haciendo que el jugador tome conciencia de la historia mediante la narrativa visual y cumpla los objetivos del juego mediante la navegación de un entorno virtual y la observación y/o interacción con objetos del juego.

¹ V. anexo en Glosario de imágenes.

El prototipo del videojuego resultante se integrará en un portfolio adecuado para poder mostrar mi trabajo tanto a estudios independientes como a estudios con proyectos de mayor envergadura.

Quiero reflejar en él mis habilidades como modelador 3D low y high poly, rigger, animador, diseñador de nivel, programador o incluso director artístico. Estas habilidades me permitirían integrarme en alguna de estas especialidades o trabajar en varias de ellas como artista generalista (*generalist artist*).

Encargarme personalmente de todos los aspectos que anteriormente menciono, me servirá para comprender cuestiones conceptuales y técnicas de cada área específica dentro del desarrollo de un videojuego, lo que me ayudará en un futuro a realizar labores de coordinación y múltiples tareas si el equipo es pequeño.

Por último, ser capaz de hacer un videojuego por mí mismo es suficiente motivación para seguir estudiando e investigando sobre este apasionante medio.

1.2. Metodología

Partiendo de una experiencia básica en el desarrollo de este tipo de proyectos, comencé por determinar el software a utilizar, buscando siempre un equilibrio entre las posibilidades que me ofrecían las distintas aplicaciones y las funcionalidades básicas que pretendía que tuviera mi proyecto, un prototipo para el nivel inicial de videojuego de aventura/acción, con los elementos y características inherentes al género, considerando la dedicación y los plazos.

Tanto Unity, que había utilizado en anteriores proyectos² como Unreal Engine, me ofrecían las características necesarias. Decidí utilizar el motor Unreal Engine 4, para explorar sus posibilidades y adquirir conocimientos que complementasen mi formación básica en los dos motores más utilizados actualmente en el desarrollo de videojuegos³. Esta decisión supuso tareas de aprendizaje y ensayo para poder realizar la programación que me permitiera alcanzar el nivel de jugabilidad y características gráficas previstas.

² V anexo Glosario de imágenes

³ ATaxonomy of Game Engines and the Tools that Drive the Industry
Marcus Toftedahl, Henrik Engström. Proceedings of DiGRA 2019

Después de este periodo de estudio y puesta en marcha empecé el primer aspecto de la preproducción, el desarrollo del aspecto visual y gráfico, comenzando por el concept art y elaborando ilustraciones tanto de personajes como de entornos que me sirvieran de referencia.

A continuación, se diseñaron los niveles, y se implementaron con geometrías previas (*placeholders*) y algunos modelados de baja poligonización para situar y escalar objetos con los que programar las físicas y las mecánicas.

Esta técnica permitió avanzar en la jugabilidad mientras se producían e integraban sucesivamente las geometrías, materiales y animaciones definitivos.

El proyecto se realizó en 11 meses compatibilizándolo con tarea académicas y laborales. El trabajo se organizó en cuatro fases: bocetos y aprendizaje de Unreal (1 mes); diseño, modelado e implementación de modelados básicos (4 meses); esqueleto, rigging, skinning y animación (3 meses) y programación y HUD (3 meses)

A continuación, se detalla el cronograma. Si bien se cumplieron estos plazos, algunos de los aspectos -como es frecuente en este tipo de proyectos- se trabajaron en conjunto o fueron rediseñados en fases posteriores tras ensayarlos.

El primer bloque ocupó un mes aproximadamente. Consistió en la evaluación de las funcionalidades del motor de juegos y la realización de dibujos conceptuales

Después de conocer el motor en funcionamiento y tener varios dibujos, pasé a diseñarlos directamente en 3D, realizando la mayor cantidad de modelados 3D posibles en la menor cantidad de tiempo, con la intención de tener modelado al personaje principal con su arma, 2 o 3 enemigos, todo el mapeado y los objetos que lo componen, creando al menos 10 elementos que no forman parte del entorno natural y la vegetación. Aproximadamente la mitad del tiempo se dedicó a modelos *low-poly* y la otra mitad a modelos *hi-poly*.

En el tercer bloque del cronograma, invertí 3 meses, que dediqué exclusivamente a la elaboración del sistema de esqueleto y *rigging* del personaje y a la realización de las animaciones y su importación al motor de juego, el cual iba configurando a la par que iba añadiendo elementos, de manera que llegados a este punto ya pude crearle un sistema de locomoción al personaje principal y moverlo por el entorno. Invertí un mes en el *rigging* y *skinning*, dos meses en las animaciones y un par de semanas en la implementación en el motor de juego.

En los siguientes 3 meses me encargué de dotar al proyecto de las características propias de un videojuego de acción / aventura, diseñando y programando las mecánicas de combate y escalada y la inteligencia artificial para los enemigos mediante blueprints⁴. También en este último periodo se diseñó e implementó el HUD. La programación, incluyendo los sistemas de batalla y escalada ocuparon dos meses y el HUD y la programación de la interacción con el entorno otro mes.

Para finalizar, un último bloque de dos meses se dedicó a depurar el proyecto, añadir el sonido y terminar los últimos detalles.

Software utilizado

Photoshop y SAI 2: Ambas aplicaciones los he utilizado para la realización de los bocetos e ilustraciones que me han servido para la ambientación y recreación del mundo en el que sitúo mi proyecto, a modo de concept art sin color, y utilizando estas ilustraciones como referencia para el posterior modelado 3D.

Maya 2018: El software que más he utilizado, de manera que me ha servido de punto de anclaje de todos los demás, ya que la mayoría de procesos pasaban por este programa en algún momento. En él, he creado los modelos 3D en baja poligonización de todos los elementos de mi proyecto, donde he realizado mayor hincapié trabajando meticulosamente los polígonos para que en conjunto funcionaran como elemento 3d de la escena según la funcionalidad que quisiera para estos, principalmente en la elaboración de personajes, teniendo muy en cuenta las zonas donde debía haber mayor cantidad de polígonos para facilitar su posterior movilidad. Todo este trabajo con la malla malla poligonal sirve para luego poder realizar en los modelos 3D de personajes un esqueleto mediante joints⁵ o “huesos” que servirán de punto de anclaje para la movilidad del personaje en su posterior animación.

Para la realización de las animaciones también he trabajado generando mis propios “rigs” o “controladores” que, enlazados con los joints, permiten mover al personaje y realizar sus animaciones.

Zbrush: Utilizado principalmente para trabajar los modelos high-poly de los elementos que lo requieran, debido a la facilidad con la que permite trabajar el detallado sobre los modelos 3D, esculpiéndolos directamente y retocando con gran nivel de detalle la maya poligonal que luego se empleará para aplicar los mapas UV a los modelos 3D definitivos.

⁴Mas información: <https://docs.unrealengine.com/en-US/Engine/Blueprints/index.html>

⁵ Término explicado en el apartado “Animación”

Substance Painter: Utilizado para la incorporación de múltiples materiales, de manera que me facilitara la labor de texturizado de algunos elementos.

Unreal Engine 4: Utilizado como motor de juego, donde he ido implementando y añadiendo los diversos elementos dentro de una escena, generando un mapa donde suceden los eventos del videojuego, así como los diseños de nivel y la composición de los mapeados, localización de objetos, enemigos, etc.

A su vez también es donde he trabajado los elementos de programación y jugabilidad del videojuego, combinando elementos de programación con animaciones para así generar todos los elementos jugables del juego: acciones del personaje, interacciones con el entorno, elementos de acción y batalla, sucesos o eventos, etc.

2. Contexto y referentes

“En la actualidad, es incuestionable que entre los fenómenos asociados a la expansión tecnológica en nuestra sociedad se encuentra el uso creciente de los videojuegos. En España, el sector de los videojuegos lidera las ventas del ocio audiovisual, habiéndose extendido esta forma de entretenimiento entre personas de diferentes edades y con distintas capacidades. Además, los videojuegos no sólo son una forma de ocio, sino que pueden constituir un buen apoyo para la divulgación de contenidos e información”⁶

La industria del videojuego cada vez se expande más y abarca a todos los estratos de la sociedad, pudiéndose encontrar diferentes perfiles de usuarios. En consecuencia, la industria del videojuego ha tenido que adaptarse a nuevos públicos dentro de un mercado en constante evolución.

Con el paso del tiempo cada vez se ven más videojuegos que, mediante tutoriales o controles simples, hacen más entendible el funcionamiento de estos y facilitan la experiencia de juego volviéndola demasiado intuitiva, de tal manera que nadie se sienta frustrado, acomodando a los jugadores hardcore, y facilitando la integración de los jugadores más casuales, como pueden ser niños o personas de avanzada edad.

Esta integración del mercado de los videojuegos en diversos sectores de la sociedad no ha impedido que hallan empresas que sigan apostando por



⁶ Buenas prácticas de accesibilidad en videojuegos, Colección Estudios Serie Dependencia N° 12014 (2012)

desarrollar videojuegos enfocados a los jugadores más experimentados, los cuales siguen demandando títulos con mejores gráficos y mayores desafíos.

Los videojuegos nos permiten experimentar sensaciones que en la vida real evitaríamos. Esto resulta útil para explicar porque a muchos jugadores les gusta experimentar el fracaso, la pérdida y el reto en los videojuegos difíciles aunque en la vida real busquemos siempre la solución fácil y que menos riesgo supone. "Un juego es un sistema en el que el jugador participa en un conflicto artificial, definido por las reglas y con un resultado cuantificable. (Salen y Zimmerman 2004, 96)"⁷

En la búsqueda de crear videojuegos que propongan retos al jugador, que te hagan pensar de que manera debes jugar para superar cierto desafío, sin que esto se convierta en una búsqueda desesperada de objetos "power up", sino que de verdad el desafío dependa plenamente de la capacidad del jugador, volver a los orígenes, cuando tu amigo no se podía pasar cierta pantalla y le decías: "¿Te lo paso?".

"En Diseño de videojuegos, equilibrio (en inglés *balance*) es el concepto y la práctica de preparar las reglas de un juego, usualmente con el objetivo de prevenir que cualquiera de sus sistemas componentes sea inefectivo o de otra manera indeseable comparado con sus pares. Un sistema desequilibrado representa recursos de desarrollo desperdiciados o por lo menos y en el peor de los casos puede minar el conjunto de reglas del juego por completo al hacer que roles o tareas importantes sean imposibles de realizar."⁸



Adventure, 1979 para la consola Atari 2600

2.1. El género de acción/aventura

Teniendo en cuenta las características con las que quiero dotar a mi proyecto, me decidí a realizar un videojuego de genero Aventura/Acción, ya que es el estilo que mejor se adapta a mi curva de aprendizaje, debido a que desde hace un tiempo he estado investigando como realizar este tipo de videojuegos, bastante completos en cuanto a mecánicas y jugabilidad, pero sin acarrear grandes dificultades extra comparado con otros géneros.

El género de Aventura/Acción es quizá el más amplio y variado en todo el mundo del videojuego y este viene a ser una fusión de ambos términos y sus respectivos géneros en videojuegos. El término "aventura" viene definido por uno de los primeros videojuegos de este género, el *Adventure* de 1979, de la

⁷ La dificultad como motivación: subgéneros definidos por el reto en el videojuego contemporáneo Mateo Terrasa Torres, Víctor Navarro Remesal (2015)

⁸ Newheiser, Mark (9 de marzo de 2009). «Playing Fair: A Look at Competition in Gaming». Strange Horizons. Archivado desde el original el 12 de marzo de 2009.



Zelda II: The Adventure of Link, 1987
Nintendo Entertainment System



Megaman X, 1993
Super Nintendo Entertainment System

consola Atari 2600, y no tiene ninguna relación con las películas de aventuras ni sus temáticas, siendo *"Colossal Cave Adventure"* el primer juego de aventuras de este tipo, lanzado por primera vez en 1976⁹. En ellos predomina la exploración del terreno, la investigación de sucesos y resolución de puzles o acertijos que el jugador debe resolver para poder continuar.

Por otro lado, el género de acción en los videojuegos viene definido por las peleas frenéticas y el uso de la violencia, aunque este último no es un requisito obligatorio. Se basan en una interacción directa contra un enemigo que desafía nuestra agilidad mental y reflejos. Ya en 1987 y con el auge de los juegos de lucha, con la recién estrenada saga de Street fighter, los juegos de aventura empezaban a caracterizarse por adquirir mecánicas de acción y combate. Otro claro ejemplo de ello es el segundo título en la saga The legend of Zelda, donde apostaron por este género.

El término Aventura/Acción apareció con el paso del tiempo de manera que se fue renovando hasta conseguir un sentido muy amplio y muchas derivaciones a su vez, pero en su término más purista vendría definido por elementos de exploración y resolución de acertijos, con la interacción en batallas.

Cabe destacar también la mecánica de recolección que utilizan muchos juegos de acción y aventura, combinando mecánicas de exploración de escenarios con recolección de objetos. Esta tendencia se podría decir que ya venía instaurada desde 1980 con Pac-Man, y que más tarde se vería en títulos tan emblemáticos como Super Mario Bros (Nes, 1985) o Sonic the Hedgehog (Sega Genesis 1991).

Como referente del género de acción cabe destacar la saga Megaman, ya que es una saga que principalmente se centra en la jugabilidad. Nada mas darle al botón de empezar partida no te muestra una intro con cinemáticas ni textos, directamente empieza la partida y te sitúa en el mapa para que empieces a moverte. Vas avanzando y se te acerca un enemigo el cual no puedes esquivar caminando, y así te indican que tienes la posibilidad de saltar. Un poco mas adelante aparece un enemigo el cual no puedes saltar, entonces descubres que tienes la posibilidad de disparar para derrotarlo. Mas adelante caes en un agujero para descubrir que puedes trepar. La saga Megaman me parece un muy buen referente en cuanto a mecánica de juego, ya que al dejar libertad al jugador sin interrumpir la acción te permite ir descubriendo las mecánicas por tu cuenta.

⁹ Para más información: wikipedia.org/wiki/Adventure_game



The Legend of Zelda Breath of the Wild, 2017 Nintendo Switch

2.2. Referentes

The legend of zelda saga

Como máximo exponente en el género de Aventura / Acción, y a su vez como referente principal en este proyecto. Es el mejor ejemplo de videojuego como obra maestra.

Gráficamente sencillo, pero a su vez increíblemente bello, siguiendo con la misma temática de antaño, pero sabiendo renovarse e implementando nuevas mecánicas que hacen la aventura mucho más atractiva. Consigue una inmersión total en su mundo y permite al jugador ser libre y a su vez guiarlo hacia el siguiente destino de la historia.

Mi mayor referencia a la hora de sumergirme en el mundo de los videojuegos, me sirve como punto de apoyo para crear un mundo desde cero, inventando personajes y props que se alejan del realismo y buscando un enfoque creativo donde todo ello parta de un mundo de fantasía que me permita tener un proceso creativo muy amplio.



Shadow of the colossus, 2005 PlayStation2

Shadow of the colossus

Otro gran ejemplo de sencillez y maestría. Una historia simple que sumerge al jugador de lleno en su mundo, dándole un objetivo principal muy claro, y dejando libertad en un vasto mundo lleno de desafíos en forma de colosus.

Las acciones del personaje como su jugabilidad, las justas y necesarias para cumplir los desafíos, son un claro ejemplo de cómo un personaje que apenas puede atacar y trepar es capaz de derrotar a los enemigos más descomunales.



Monster Hunter Ultimate, 2011 WiiU

Monster Hunter saga

Perfecto referente por lo que corresponde a elementos de acción y exploración, pero con toques de rol que también me sirven de ejemplo de otro tipo de género que no he querido aplicar a mi proyecto, debido a la complejidad de trabajo relacionado con objetos que aumentan los "Stats". Pero en su totalidad me sirve como referente en lo que respecta a buscar y enfrentar grandes enemigos.



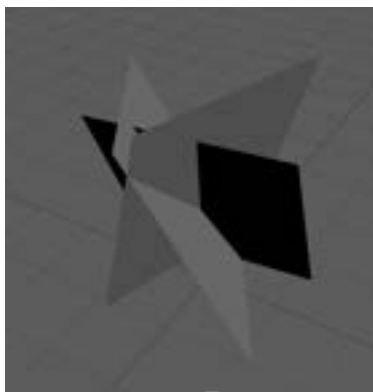
3. Concepto y desarrollo del proyecto

3.1. Concept art / dibujos previos

El comienzo de este trabajo parte de un estudio e investigación sobre el concept art, con el cual he querido buscar una estética bastante peculiar que, pese a no alcanzar un gran nivel de detalle, atiende a una estética manga / anime que busca cierto realismo dentro de la fantasía. Todos los props han sido diseñados de manera que atienden a una estética realista, tanto en formas como en texturas, pero a su vez buscando que no fuera exagerado. Implementando tintas planas en texturas, buscando acercarme a una estética realista, pero trabajándola desde la sencillez de las formas.

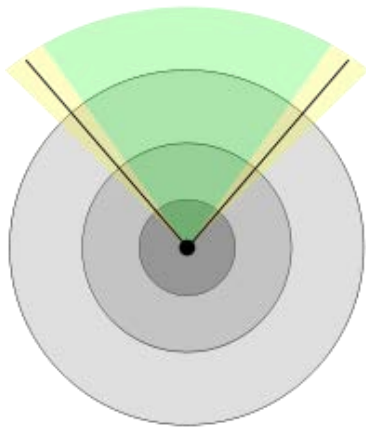
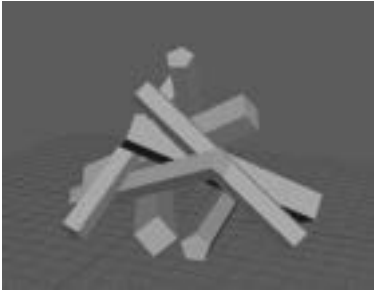
3.2. Modelado del terreno

Todo el terreno ha sido creado mediante la herramienta de “landscape” del motor de juego modelándolo manualmente y posteriormente se han añadido los elementos orgánicos de decorado mediante la herramienta “foliage”.



3.2.1. Vegetación y Rocas

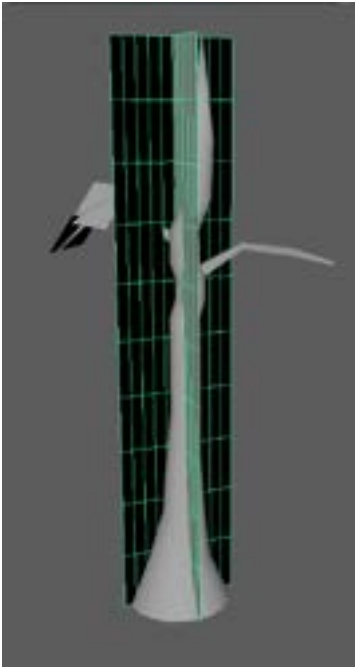
La creación de los elementos vegetales ha sido muy diversa y no siempre me han dado un buen resultado, debido a que una gran acumulación de elementos vegetales, que además son animados con efecto de viento, gastan muchos recursos en la plataforma, por lo que



fueron configurados mediante LODs (Load on distance) y culling distances, de tal forma que estos no eran generados en el render si no aparecen en nuestro campo de visión, y de manera que cuando nos vamos acercando a ellos, las capas de LOD van generando los modelos de menor a mayor calidad progresivamente, para así conseguir que no gasten tantos recursos. En esta imagen podemos comprobar de una manera mas gráfica el funcionamiento de esta configuración del terreno, de forma que los elementos que se encuentran en el círculo oscuro del rango de visión son cargados con el modelo de mejor poligonización, mientras que las zonas de rango de visión más alejadas, reflejadas en la imagen en tonos más claros, se estarían cargando los modelos de poligonización más baja progresivamente según su distancia.

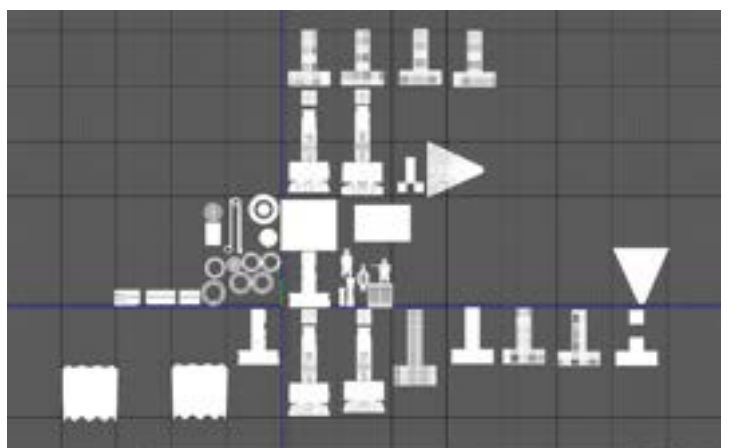
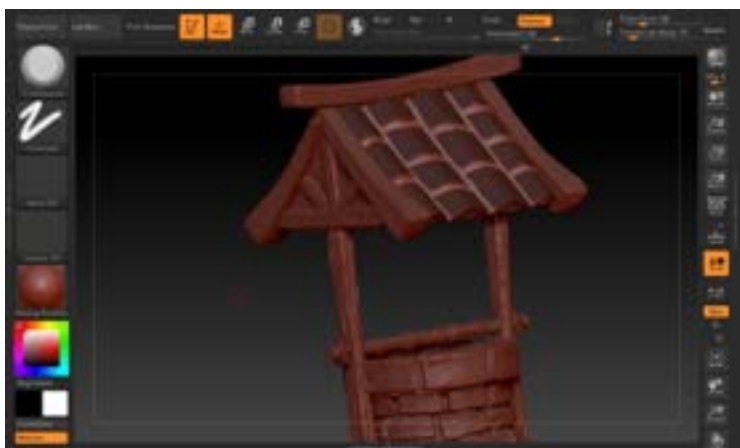


Ejemplos de la ambientación , primero con LODs y cull distance sin configurar, y segundo con estos configurados a gran proximidad.



3.2.2. Otros elementos del terreno

La mayoría de elementos han sido creados en low-poly debido a la falta de tiempo y querer enfocarme en otros aspectos, pero a su vez he querido realizar pruebas con algunos de estos elementos, generándoles los modelos high poly y su despliegue de malla poligonal con el que poder realizarles los mapas UV. Cada elemento ha sido texturizado con mapa de color base y normal map, como mínimo, para dar una mayor sensación de relieve a los objetos.

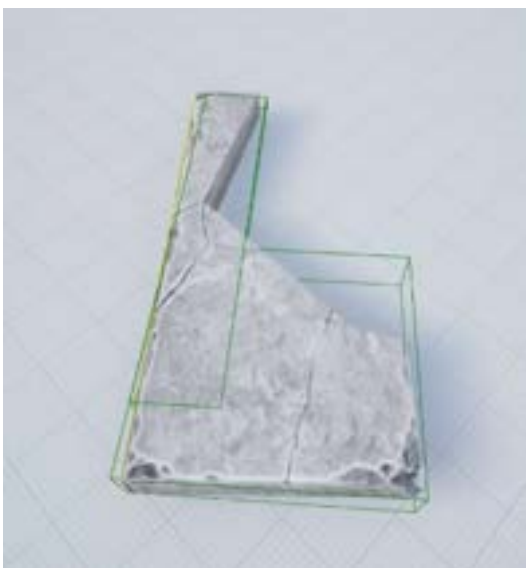


Utilizando la herramienta Mask para los pliegues y mapeado UV del pozo en Zbrush.

3.2.3. Físicas y colisiones

El sistema de colisiones ha sido integrado de manera que se diferencian dos tipos de colisiones en objetos según se puede interactuar con ellos. Están los objetos que solamente colisionan con el personaje, dichas colisiones han sido generadas de manera automática por el motor de juego, y después están los elementos que colisionan con el personaje y le permite treparlos, diferenciándolos de los otros elementos de colisión en que en estos he tenido que realizar un sistema de colisiones personalizado para que se adaptara a la geometría del objeto, permitiendo que el personaje lo trepara de manera correcta.

Llegados a este punto, la acumulación de elementos orgánicos en funcionamiento sobre el mapeado, con sus respectivas animaciones y efectos, hacían que el motor de juego bajase su rendimiento, debido a la gran carga de polígonos que conlleva generar tantos elementos. Para solucionarlo se utiliza un sistema de optimización que permite no renderizar elementos que quedan fuera de nuestro punto de vista, o renderizarlos a una resolución menos según a la distancia que nos encontremos de estos.



3.3. Interfaz y controles

Tras diseñar el entorno, el siguiente paso es implementar las mecánicas jugables que va a poder hacer el jugador sobre este. En el diseño de un videojuego, es muy importante tener en cuenta los movimientos que nuestro personaje puede realizar, y esto debe darse de manera intuitiva y en relación a lo que los elementos del entorno nos permiten hacer, proporcionando unos controles que a su vez también sean intuitivos, y permitiendo al jugador descubrir si el personaje puede correr, saltar, colgarse de salientes o deslizarse sobre una rampa.

El personaje principal de mi proyecto tiene las siguientes capacidades:

Correr: Permitiendo al personaje moverse más rápido, pero a su vez gastando la barra de estamina. Se usa manteniendo la tecla “Shift”

Saltar: Permite al personaje llegar a salientes, saltar espacios vacíos e incluso esquivar ataques enemigos. Cabe destacar que se salta mucho más mientras se corre. Se salta pulsando la barra de “espacio”

Colgarse de salientes: Esta habilidad permite al personaje evitar caídas y poder bajar sin problemas por precipicios mediante salientes. Si la parte superior del personaje se acerca a un saliente, este se coge directamente. Pulsando la flecha hacia abajo el personaje se descuelga.

Trepar salientes: Cuando el personaje se encuentra colgado, pulsando la flecha hacia arriba este trepa para subirse a la superficie de donde se encuentra colgado.

Ataque débil: Utilizando el clic izquierdo el personaje ataca dos veces rápidas y consecutivas, pero el daño que quita es bastante bajo.

Ataque fuerte: Utilizando la tecla E el personaje realiza un ataque lento pero devastador que, aunque deja expuesta la defensa del personaje, lo compensa con mucha fuerza de ataque.

Cubrirse: Con clic derecho nuestro personaje se defiende de golpes enemigos, manteniendo una posición de guardia y evitando que su salud disminuya.

Esquivar: Cuando el personaje se esta cubriendo, podemos utilizar las flechas de movimiento para realizar una esquiva hacia cualquier dirección. Esto me permite darle un enfoque más profundo al sistema de batalla.

Ataque a distancia (bomba de luz): Utilizando la tecla Q el personaje realiza un ataque de larga distancia que genera una bomba de luz, la cual podemos

lanzar soltando la tecla Q, o también la podemos dejar caer delante nuestra, si antes de sacar la bomba pulsando la tecla Q mantenemos pulsado el botón derecho del ratón para que nuestro personaje se cubra.

Leer / interactuar: Utilizando la tecla R el personaje puede leer carteles.

3.4 Diseño de nivel y mecánicas

Al empezar el nivel nos encontramos con una amplia zona boscosa, la “zona de calma” donde ningún enemigo te puede atacar, de manera que el jugador se pueda familiarizar con los controles. El nivel está diseñado de manera que cuanto más nos acercamos a la zona de las ruinas, más enemigos nos encontramos, aumentando la dificultad progresivamente. Antes de llegar a la zona final del templo, quise hacer un pequeño guiño a uno de los referentes nombrados en el apartado visto previamente, como es el caso de MegamanX, donde dejan caer al jugador a un agujero del que no puede salir de manera que tiene que descubrir que puede trepar. De igual manera elaboré una zona de madera vieja que al posarte sobre ella durante más de dos segundos esta se rompe, dejándote caer a un agujero donde la única manera de salir es trepando.

Al avanzar con el personaje desde la zona del principio, se alcanza una zona de ruinas, diseñada con un camino marcado por salientes por los que el personaje debe trepar por un diseño de nivel enfocado a la exploración, saltando entre rocas, y buscando el camino por donde pueda conseguir más monedas, teniendo en cuenta que durante esta travesía también podrá encontrarse con enemigos.

La zona de las ruinas ha sido diseñada en su mayoría mediante copias (“instances”) de modelados base realizados previamente, debido a la repetición de sus elementos. En este nivel he conseguido llevar a cabo un auténtico desafío, ya que no es fácil llegar al final sin ser derrotado por los enemigos, o sin caerse escalando alguna repisa. El nivel ha sido diseñado de tal manera que, si el personaje se cae mientras está escalando la zona del templo, caerá directamente a la zona baja del templo donde hay mayor número de enemigos, pero también decir que los enemigos más fuertes están en la parte de arriba del nivel llegando al final.



3.5. Hud

El HUD(Head-Up Display) de mi proyecto ha sido diseñado de tal manera que sea intuitivo y permita al jugador conocer la información necesaria para su partida, sin que este llegase a ser molesto. Aunque también es cierto que me hubiera gustado integrar más características en el HUD ya que esto también significa que el juego tiene diversos elementos jugables e importantes que son necesarios tener a simple vista, creo que he aportado a mi proyecto los elementos necesarios que lo hacen atractivo de cara al jugador, dando una experiencia de juego completa e intuitiva.

“El ojo humano suele mirar primero la esquina superior izquierda, ¿por qué? porque empezamos a leer por ese sitio. Esto cambia en culturas donde se lee de derecha a izquierda. Por eso normalmente se posiciona la barra de vida arriba a la izquierda.

La parte superior e inferior central suele ser un lugar idóneo para textos, información de objetos o para que aparezca por unos instantes mi inventario de objetos.

Y, efectivamente, la parte en la que nos solemos fijar menos es la inferior izquierda. Allí solemos poner las cosas menos importantes.



Cada uno puede colocar los elementos donde mejor le convenga para su juego, pero tener claro qué sitios suele ver más el ojo humano es importante para diseñar el HUD con criterio”¹⁰

-Barra de Estamina

La barra de estamina ha sido diseñada de manera muy sencilla mediante un rectángulo pintado de verde, y programado de tal manera que esta barra se vacíe cuando el personaje corra, para poner un límite de tiempo en el sprint. Esta barra también atiende a la idea de poder utilizarla también en la implementación de magias o hechizos del personaje principal, que también gastarían la barra. También ha sido programada de manera que se regenera de manera automática al pasar un segundo sin gastarla.

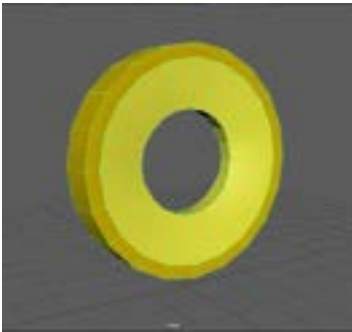
-Vidas

La barra de salud, o corazones, han sido diseñadas de igual manera que la barra de estamina, pero diseñando y colocando sobre esta los corazones y programándola de tal manera que cada punto de vida corresponde a medio corazón, y este se vacía cuando el personaje sufre daño.

Para el diseño de los corazones se han creado la imagen con todos los corazones llenos, y otra imagen igual, pero con los corazones vacíos, y estas se van entremezclando según la barra de vida se consuma.



¹⁰ Pagina: www.gametopia.es, en su artículo sobre como diseñar un hud de videojuego



-Recolección de monedas

Esta interacción con las monedas del terreno ha sido elaborada de manera que hubiera un objetivo más para el jugador, el de recolectar la mayor cantidad de monedas posible a lo largo de la partida, de tal manera que se van acumulando en la parte inferior izquierda de la pantalla.



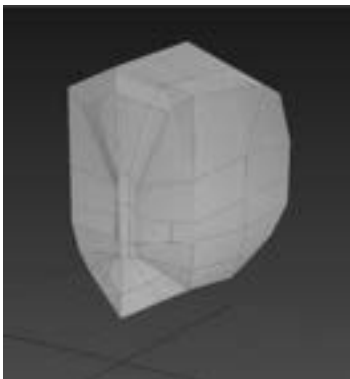
El objetivo con estas monedas es conseguir 50 de ellas a lo largo del nivel sin que los enemigos acaben con tu vida. En muchos de los casos sirven para guiar al jugador por el camino correcto del diseño de nivel. Estas monedas no tienen ninguna utilidad mas allá de su recolección, ya que fueron creadas como un elemento más del diseño de nivel y de manera que me sirvieran en un futuro para que en la partida pudieran ser gastadas en mejoras u objetos para el personaje principal.

3.6. Personajes

3.6.1. Modelado

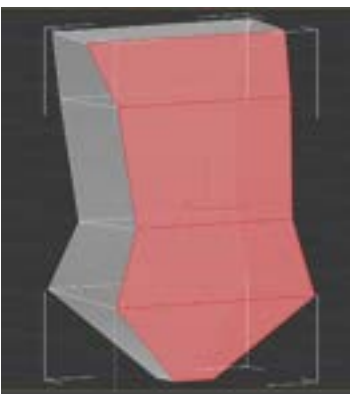
3.6.1.1. Personaje principal

Modelador Low Poly



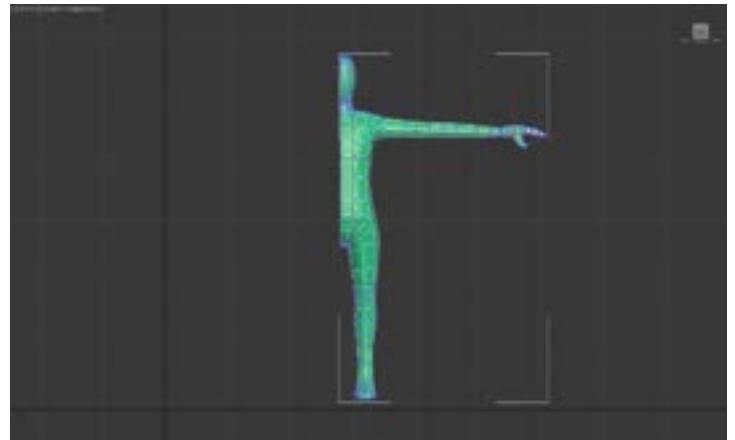
Como Low Poly, nos referimos a las mallas poligonales en 3D que tienen un numero relativamente pequeño de polígonos. “Las mallas de baja poligonización se utilizan en aplicaciones 3D en tiempo real, como es el caso de los videojuegos en contraposición con las películas animadas por ejemplo, y son un factor importante para optimizar el rendimiento y para disminuir los tiempos de renderizado (es decir, aumentar la tasa de fotogramas), pero puede acabar dando una apariencia indeseable a los gráficos resultantes.”¹¹

“Las técnicas de procesamiento de gráficos 3D, como el mapeo de normales y de relieve, se han diseñado para compensar la disminución de los polígonos al hacer que un objeto con bajo número de polígonos parezca contener mas detalle de los que contiene y esto se hace alterando el sombreado de los polígonos.”¹²



¹¹ Dariush Derakhshani, Randi Lorene Munn (2008). *Introducing 3ds Max 2008*. John Wiley and Sons. p. 240. Retrieved 12 July 2011.

¹² Thompson. *Game design course: principles, practice, and techniques*. Retrieved 16 July 2015.



Para la realización de un modelado correcto hay que tener en cuenta que la superficie del modelo mantenga una coherencia formal que permita la aplicación de deformidades propias del posado o la animación. Esto se traduce mediante una buena topología que permite un correcto despliegue de la malla según las coordenadas UV para la aplicación posterior de los distintos mapas. La cabeza y las manos son las partes más densas, ya que deben soportar deformaciones más intensas, y por lo tanto, deben tener más cantidad de polígonos.

Empezaremos creando la cabeza del personaje, partiendo desde un cubo al que vamos a crear segmentos que delimiten las partes más importantes o de referencia en nuestro personaje, como son los ojos, la nariz y la boca. Es importante crear intersecciones en las partes de la cara donde van a haber más detalles, para así poder ir añadiendo vértices y segmentos que marquen las zonas de detalle. Nos ayudamos principalmente de herramientas como son “Connect” para conectar los vértices necesarios, así como “Inset” que nos permite generar geometrías a partir de otras ya existentes. Otras herramientas muy útiles para generar volúmenes a partir de geometrías son “Extrude” y “Bevel” entre otras.



Modelado High Poly

Una vez hemos modelado, detallado y pintado en Zbrush podemos sacar los UV Maps de nuestro modelo, para ello iremos al menú principal: Zplugin/ UV Master. Donde pulsaremos sobre la opción “Work On Clone” para que nos genere una copia en low poly de nuestro modelo en blanco, para poder trabajarle los Uvs y transferirlos al modelo original.

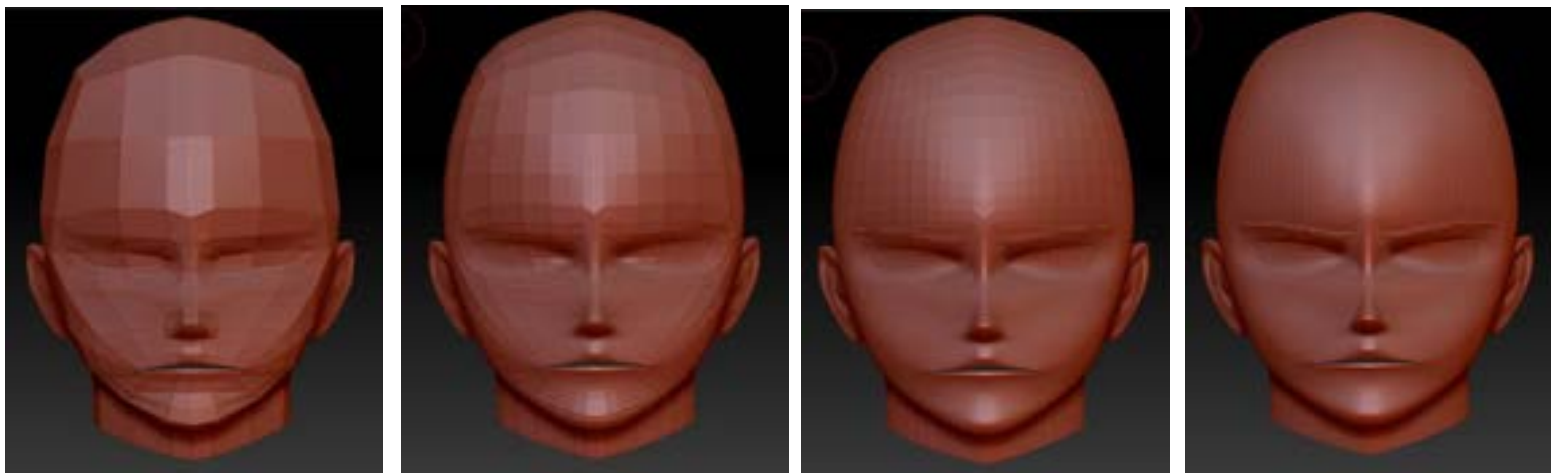
Activamos la opción “Enable Control Painting” y esto activara las tres opciones “Protect” para pintar de rojo las partes importantes de nuestro modelo que no queremos que se vean afectadas, “Attract” nos sirve para pintar de azul la zona por donde debería cortarse el mapaUV, y “Erase” por si queremos corregir algo.

Aplicamos el “Unwrap” y nos generará el mapa dejando nuestro modelo blanco de nuevo.

Aquí muestro la progresión en la subdivisión de la cabeza del personaje: donde la primera imagen corresponde al modelo low poly y tiene 498 poligonos, la segunda 1950, la tercera 7800 y la última, que corresponde al modelo high poly tiene 31200.

También podemos pulsar el botón “CheckSeams” para visualizar mediante líneas amarillas las zonas donde se ha realizado el corte para el mapaUV.

Despues debemos pulsar la opción “Copy UVs” en la misma pestaña de “UV Master” para asi poder transferir la información de nuestra copia al modelo original. Al pulsarlo vamos a la ventana de “Tools” y activamos nuestro modelo original, para ir de nuevo a la pestaña de “UV Master” y aplicarle la opción “Paste UVs”.



Exportando los mapas de textura, desplazamiento y normales:



A partir de aquí ya podemos sacar los distintos mapas del modelo, empezando por el de textura, ubicándonos en la pestaña lateral de Tools y pulsando sobre “Texture Maps”, para luego pulsar “New From Polypaint”, generando la textura de nuestra cabeza.

Para exportar la textura pulsamos sobre “Clone Txtr” dentro de “Texture Map”. Esto nos clonara nuestra textura a nuestro menú de herramientas, en la pestaña “Texture”, donde podremos pulsar “Export”. También es importante saber que a la hora de exportar los mapas estos pueden estar invertidos, por lo que antes de exportarlos debemos sobre el botón que dice “Flip V”. Para exportar el mapa de desplazamiento es necesario bajar el nivel de subdivisión al mínimo, luego nos ubicamos en el menú lateral de tools sobre la pestaña “Displacement Map” y comprobamos que están seleccionadas las opciones “Adaptive” y “SmoothUV”. Pulsamos sobre “Create And Export” y listo.



Para terminar, con el mapa de normales seguimos la misma mecánica, ubicándonos en la pestaña lateral llamada “Normal Map” y nos aseguramos de activar los botones Tangent, Adaptive y SmoothUV. Al igual que en el mapa anterior, tiene que estar en la subdivisión más baja. Pulsamos sobre “create Normal Map” y cuando termina seguimos el mismo proceso que con el mapa de texturas, pulsando “CloneNM” para que nos lleve el mapa a la barra de herramientas derecha y de ahí poder exportar el mapa de normales.

Quise probar las físicas sobre ropa bastante detallada que realicé, pero el problema viene con la cantidad de polígonos que utilizan estas prendas para conseguir una movilidad totalmente fluida, por lo que tuve que decantarme por hacerle al personaje principal un ropaje más sencillo y que no gastara tantos recursos en movimiento.





3.6.1.2 Enemigos

Este enemigo lo realice de forma muy sencilla, ya que se trata del enemigo más básico, y al tratarse de un personaje antropomórfico le realice el mismo proceso que paso a detallar en el apartado de “Personaje principal”.

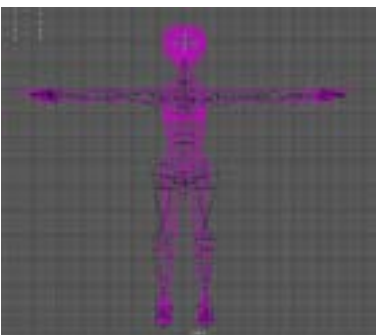
Realicé la programación para este enemigo de manera que se encontrase dentro de un área llamada Navigational Mesh¹³ y que fuera como el área de batalla donde, en el momento que el personaje principal entrase en esta área, el enemigo se dirige hacia él eligiendo la mejor dirección. También le programé un modo de alerta en el que, los enemigos se encuentran patrullando y, al oír o ver al personaje se dirigen hacia él. También lo programé de tal manera que calculase la distancia con el personaje principal y así también pudiese realizar un ataque a distancia. Pero esta habilidad no se la integré a este enemigo en la demo final, guardándomela para poder utilizarla en otros.



También le realice el proceso de mapeado UV y un riggeado personalizado con el que poder realizar las animaciones necesarias que debe tener todo enemigo en un videojuego.



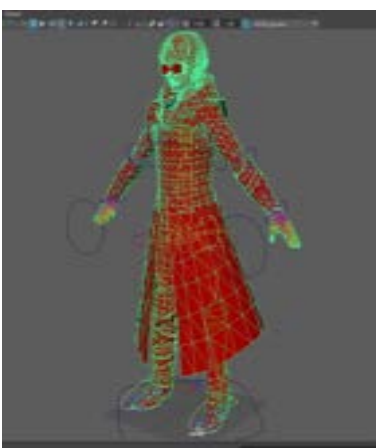
También elabore otro enemigo mediante el modelo base que utilice con el personaje principal, de manera que este enemigo pudiera tener las mismas animaciones y funcionalidades que el personaje principal.



3.6.2. Rigging y skinning

El **rigging** es el proceso de crear un sistema de controles digitales y agregárselos a un modelo 3D para que así pueda ser animado fácil y eficientemente. Este es un paso crucial dentro del proceso de la creación de una animación 3D. Como dijo Johan Arzuza "Un Rigger es como un creador de marionetas, pero en este caso como son modelos digitales, podríamos decir entonces marionetas digitales"¹⁴

El rigging es un proceso fundamental en cualquier producción de animación, cine o videojuegos, ya que se encarga de dotar de “esqueleto” y articular cualquier elemento que vaya a ser animado. En la realización de este



¹³ Definición de Navigational Mesh en el glosario de términos.

¹⁴ Arzuza, Johan. ¿Qué es el rigging? (2011). www.artzuza.com

proyecto he producido un mínimo de dos riggeados distintos para tres tipos de elementos animados como son el personaje principal y los dos enemigos, ayudándome a comprender mediante tres ejemplos anatómicamente diferentes, sus distintas necesidades a la hora de animarlos y permitiéndome entender el funcionamiento de cualquier tipo de riggeado anatómico.

Para conseguir un rig adecuado para su posterior animación se han de tener en cuenta diversos factores a la hora de trabajar sobre un modelo:

-El personaje antropomórfico debe de estar en pose de T, es decir, con los brazos separados del cuerpo estirados horizontalmente, con las manos mirando hacia abajo, los dedos separados entre ellos, y las piernas separadas con los pies apuntando al frente.

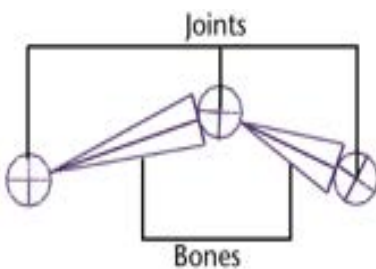
-Es necesario haber realizado una correcta topología del modelo para que haya suficientes divisiones en las zonas de articulación donde van a haber más deformidades.

- Agrupar todos los elementos que componen al modelo en una sola malla mediante la herramienta Mesh/Combine. Los únicos elementos que no se combinan con el resto de la malla son los ojos, ya que son controlados de manera independiente para poder animarlos por separado.

-Por último, hay que congelar las transformaciones en el modelo, restaurando sus valores de traslación y rotación a 0 de manera que no de problemas de coordenadas a la hora de animarlo (mediante: Modify/Freeze Transformations).

Los elementos animados son previamente riggeados mediante la elaboración de joints, que son los puntos de conexión entre los “huesos” y a su vez hacen de articulación, pudiendo limitar o incluso bloquear el movimiento de estos en algún eje concreto o hacia alguna dirección, como son el caso de doblar la rodilla hacia delante o girarla verticalmente. Cabe destacar que se puede unir un joint con varios otros pudiendo generar bifurcaciones. También decir que todos los componentes que forman el rig no son renderizados ya que únicamente se usan para generar las animaciones. Los huesos no tienen una presencia física en la escena ya que solamente son señales visuales que nos muestran las relaciones entre los joints.

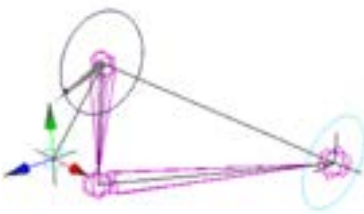
La intención es generar un esqueleto y que este nos sirva como la jerarquía de articulaciones o joints conectados entre sí, siendo capaces de generar una movilidad en nuestro modelo. “Parenting” es el término utilizado cuando se coloca una articulación por debajo de otra dentro de una jerarquía, de modo que se convierte en un elemento secundario de la articulación que se





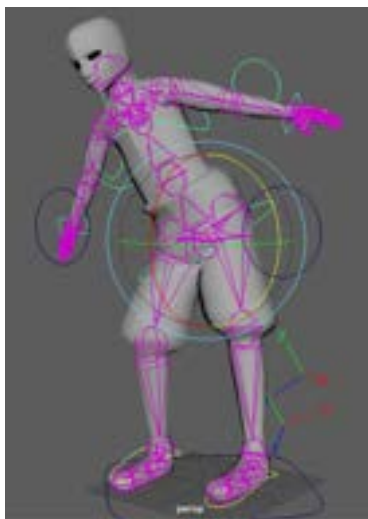
encuentra por encima (su principal) en la jerarquía. Una articulación actúa como nodo principal o padre sobre cualquier otra articulación que ocurra en la jerarquía por debajo de ella, afectándole en su punto de giro rotativo desde la base. Así pues, conseguimos que este esqueleto actúe de la manera más parecida a un esqueleto humano, donde su articulación padre dentro de la jerarquía sería la de la cadera. Otros ejemplos los encontramos en animales de dos y cuatro patas, pero los esqueletos también son útiles para animar otros elementos orgánicos con diferentes modelos jerárquicos, como son los ojos y gestos faciales, la cola de un animal, un tentáculo o una serpiente.

Al tratarse de un personaje bípedo, al igual que con el enemigo 1, se puede realizar el esqueleto de este mediante joints partiendo de una mitad de este y generando posteriormente unos mirrors simétricos que generan el mismo grupo de joints en la otra mitad del personaje.



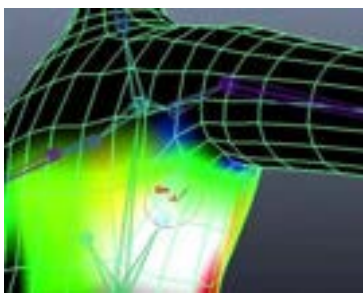
El “parenting” simple es una relación permanente entre el padre y el hijo dentro de la jerarquía. Se puede animar al hijo independientemente, pero este se verá afectado por los movimientos de rotación, traslación o escalado del padre. Si seleccionamos el joint de la muñeca, se verán seleccionados también el de la mano y los de los dedos, y si movemos el joint de la muñeca también moveremos todos los demás.

Para poder animar ciertas partes del cuerpo con facilidad la técnica utilizada es la del “parent constraint”, la cual se diferencia del “parenting” simple utilizado anteriormente en el esqueleto en que esta acción de emparentar te permite realizar uniones entre joints dentro de la jerarquía sin importar su posición dentro de esta, dotando a estos joints de ciertas libertades con respecto a los joints superiores y facilitando la movilidad de partes del cuerpo como son las rodillas, los codos o la espina dorsal y evitando futuros problemas.



El “parenting constraint” (IK Handle) es una técnica de animación que consiste en crear una relación jerárquica entre dos o más joints que podrán ser controlados de forma independiente y a su vez, podrán ser restringidas ciertos movimientos de traslación y rotación para conseguir el resultado deseado. Por ejemplo, si realizamos una rotación en la cadera podemos comprobar como se producen los movimientos en los joints afectados de una manera uniforme y natural.

El siguiente paso para la realización de un correcto rig es unir la malla poligonal que forma nuestro personaje con el esqueleto a medida que le hemos creado, de manera que, la malla cubra el esqueleto, manteniendo a este en su interior. La técnica utilizada para esto es “Bind Skin” (Skin/ Bind skin), que consiste en la aplicación de unos deformadores sobre la malla



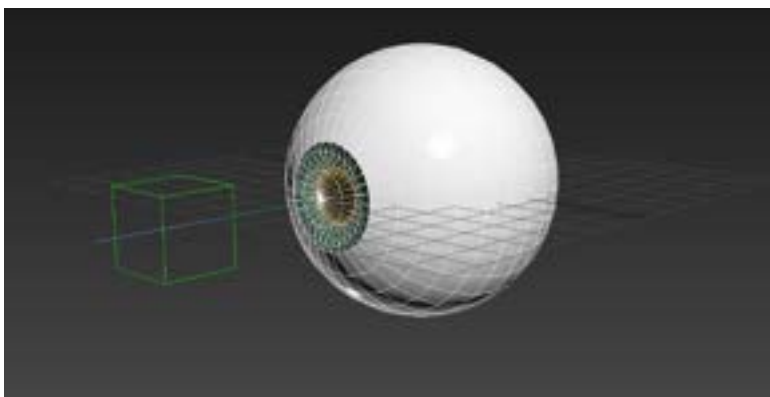
poligonal que permiten que esta sea movida por el esqueleto que contiene, gracias al efecto envolvente que la malla hace sobre los huesos. Los vértices de la malla se vinculan con los joints del esqueleto permitiendo arrastrar la geometría de la malla en los movimientos producidos por el esqueleto.

Para terminar de ajustar los detalles en las zonas que causen problemas en el deformado o torsión, existe una herramienta, el “Paint skin weights”, para pintar sobre la malla poligonal del modelo el peso de la influencia de cada hueso sobre los vértices de la maya sobre los que actúa.

Una vez ajustado todo, el último paso es el de crear los controladores, que no son más que figuras geométricas que colocamos fuera del modelo y nos ayudan a mover el esqueleto junto con la malla ya enlazada sin necesidad de trabajar sobre los joints directamente. Estos controladores se sitúan de manera que sea más fácil trabajar las posteriores animaciones, partiendo todos ellos de un controlador principal que se coloca en la base del modelo y nos sirve para moverlo en conjunto, y a su vez también hace de padre en la jerarquía de controladores que generaremos de igual manera que hicimos con los joints.



También realicé pruebas para que el personaje pudiera seguir con la mirada a un cubo y así poder animarle también los ojos, pero en la versión final descarté esta idea porque no iba a aportar nada relevante dentro del proyecto y preferí centrarme en otros aspectos. La configuración necesaria para que el ojo mire a un punto fijo se obtiene utilizando el Aim constraint, mediante la cual el ojo estará orientado para mirar a un controlador o dummy creado previamente y que utilizamos para mover el ojo en la dirección deseada sin necesidad de rotarlo directamente.



3.6.3. Animaciones

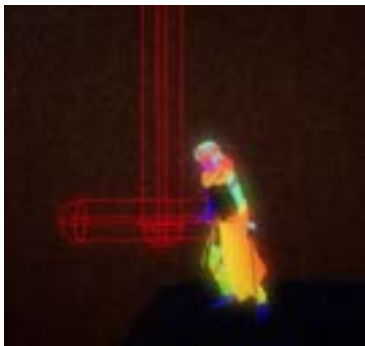


Todas las animaciones de mi proyecto han sido elaboradas personalmente, utilizando además un sistema de huesos y controladores creados exclusivamente para cada personaje.

Se han utilizado animaciones con poses base como referencia, sobre las que partirían el resto de animaciones, en un ciclo programado de manera que se enlacen unas con otras desde el mismo punto de partida, permitiendo una movilidad fluida y coordinada.



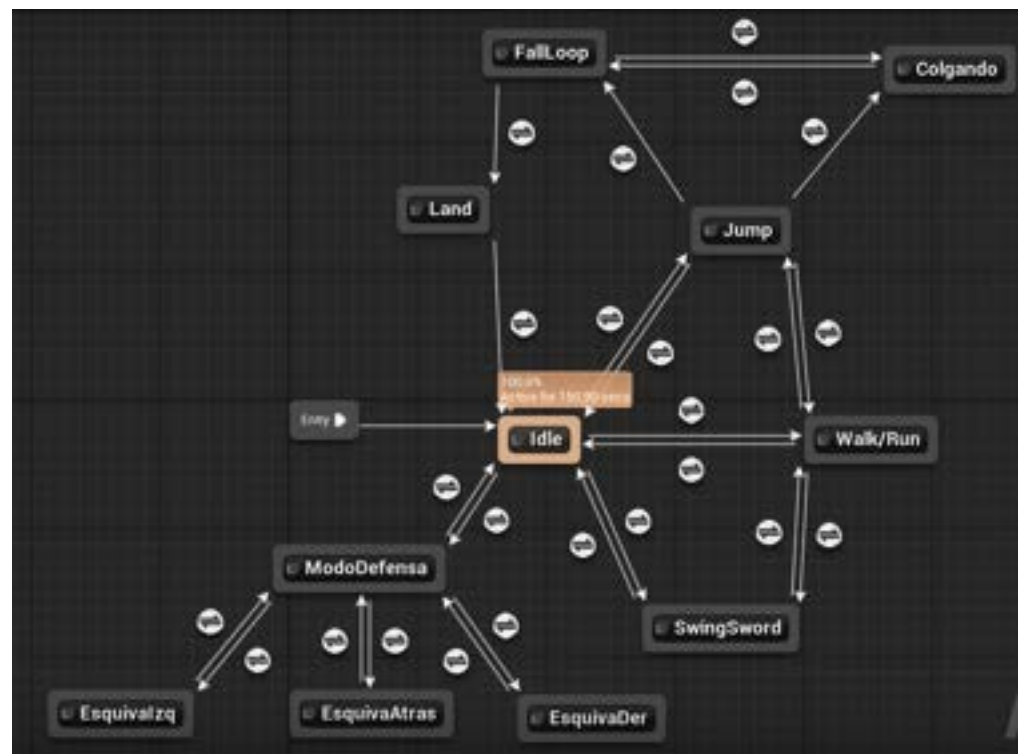
3.6.4. Programación y Jugabilidad



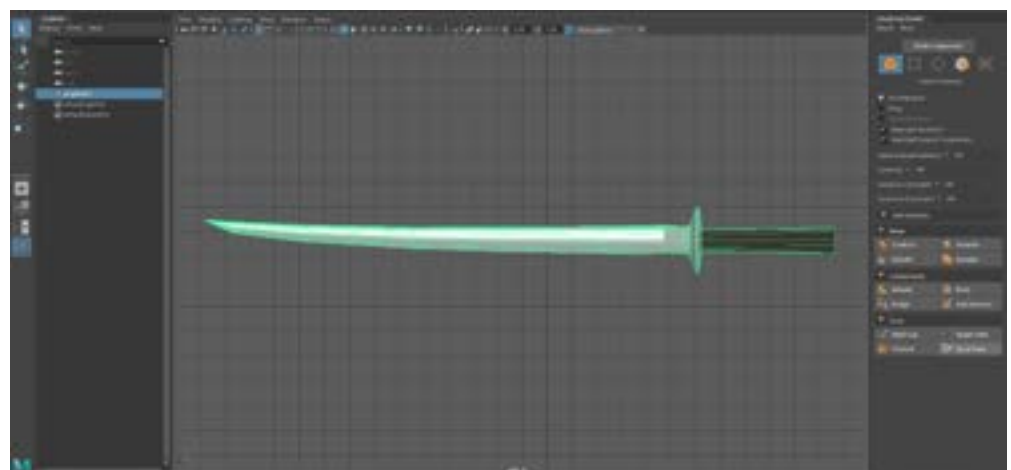
La programación creada para este proyecto ha sido enfocada a realizar acciones de combate y exploración, de manera que se pudieran implementar las animaciones previamente creadas mediante rigging, y poder utilizarlas en mecánicas jugables, tanto de combate, como son los movimientos del personaje de esquivar, atacar, bloquear, como en mecánicas de exploración, como es el caso de trepar cornisas. Esta última mecánica, la de escalar, fue posiblemente la que mas trabajo me ha dado, y la que posiblemente peor resultado haya dado y más le quede por depurar, ya que me las he tenido que ingeniar para inventarlo sin tener gran experiencia en mecánicas de juego. La mejor manera de conseguir hacerlo fue aplicándole al personaje principal un “tracer” o tiralíneas tanto frontal como vertical, que lo que hacen es detectar si el personaje tiene un objeto delante y calcular la distancia y la altura a la que se encuentran de este, de manera que estos “tracer” pueden interactuar con ciertas repisas o bordes, previamente programados, y al detectar al personaje le permiten realizar la interacción de escalar, siempre y cuando salte y alcance la distancia máxima de ambos tracer (el personaje choca contra la repisa). En este momento se reproduce la animación correspondiente, y deja al personaje colgado de la repisa, de manera que si pulsamos hacia abajo se deja caer, y si pulsamos hacia arriba se reproduce la animación correspondiente y se impulsa al personaje permitiéndole trepar el obstáculo.



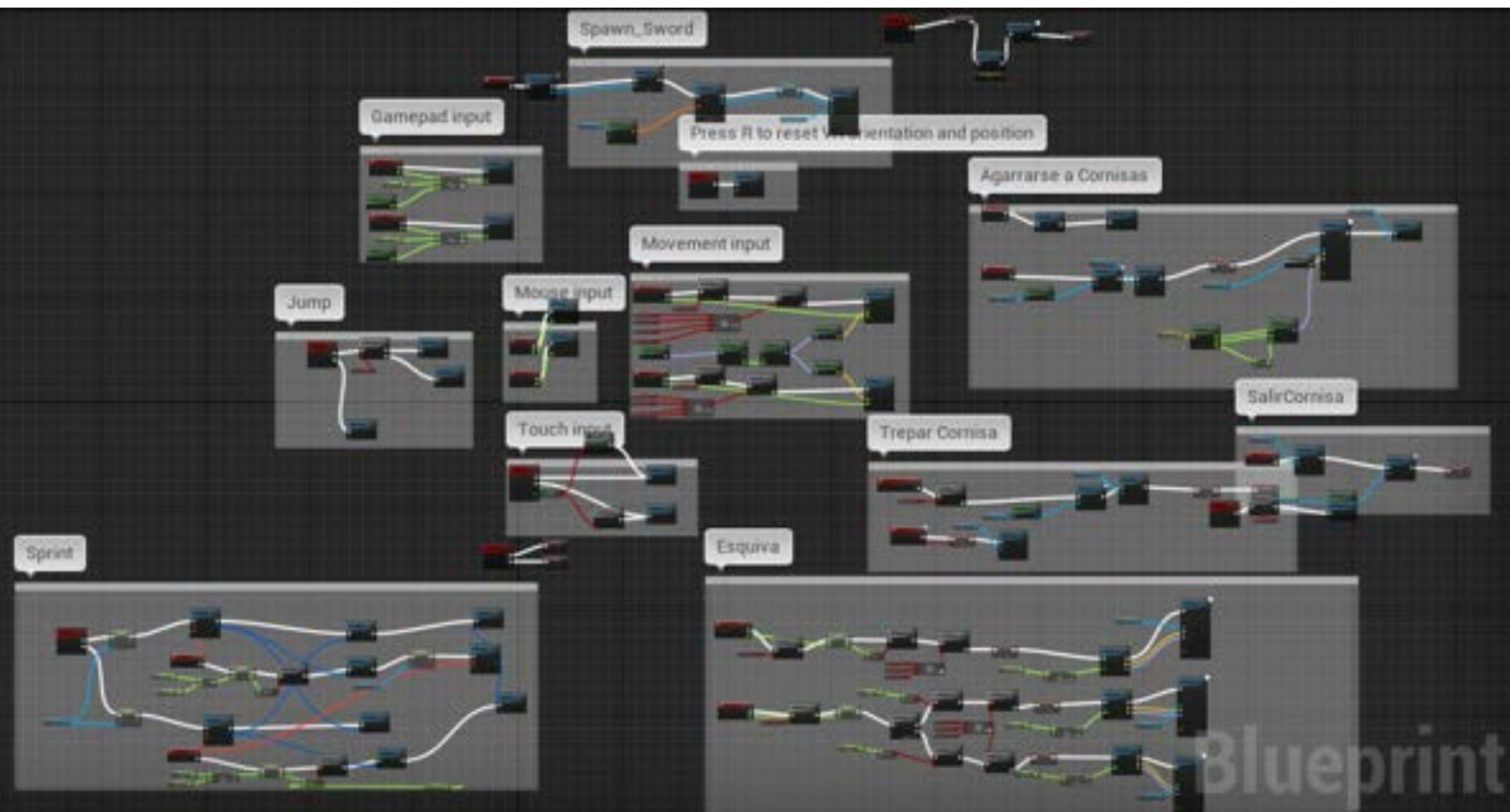
También se le ha añadido al personaje la capacidad de atacar a distancia lanzando una bomba de luz que explota impulsando a los enemigos. Esta idea fue elaborada inspirándome en armas secundarias de otros videojuegos como son las bombas de la saga “The Legend Of Zelda”.



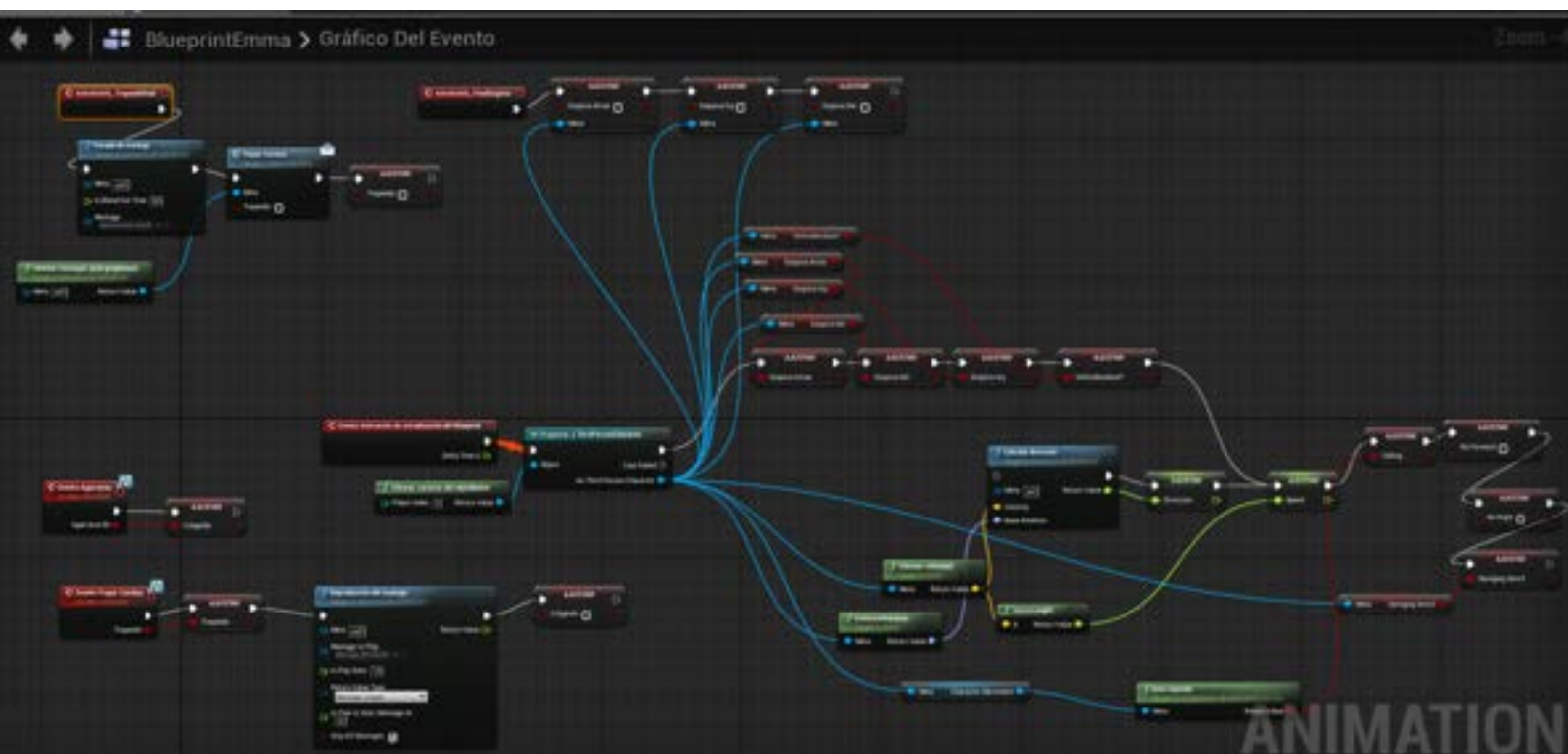
La espada ha sido integrada al esqueleto del personaje mediante un socket, que enlaza el mango de la espada con la mano del personaje, y se ha programado de tal manera que la espada aparece con el esqueleto al empezar la partida. También tiene integrado un sistema de colisiones con el que poder detectar cuando entra en contacto con los enemigos, de manera que la espada le aplica daño, siempre y cuando el personaje principal se encuentre atacando.



Blueprint general del personaje principal



Blueprint de animaciones del personaje principal:



3.7. Futuras mejoras previstas.

Elaborar mas objetos o armas que puedan ser obtenidos mediante monedas.

Realizar una prueba saltando desde un acantilado y cogiéndose a la espalda de un gigante, de manera que haya que escalarlo hasta llegar a su cabeza para poder derrotarlo.

Terminar de detallar y perfeccionar la zona de las ruinas.

Utilizar este proyecto como base del modo exploración para un rpg táctico, donde se utilicen las monedas recogidas en este para contratar y mejorar personajes, proyecto que ya esta en desarrollo.

4. Conclusiones

Una vez terminado este trabajo, me siento satisfecho con el resultado, en tanto que he conseguido diseñar y desarrollar el prototipo implementando las mecánicas de exploración y de peleas en el motor Unreal Engine 4. El prototipo es funcional y cumple (y en algún aspecto supera) las características visuales y de jugabilidad que pretendía en un principio.

En el proceso, he elaborado el diseño, el modelado 3D, el *rigging*, la animación, la programación y la música, aprendiendo y practicando conceptos y técnicas de todas estas áreas, aprendizaje que me ayuda a desarrollar un perfil profesional adecuado a mis objetivos. Todo ello integrando conocimientos adquiridos en la realización del Grado en BBAA y de manera autónoma.

El proyecto me ha reportado el aprendizaje necesario para abordar proyectos más complejos, en los que estoy aplicando desde el principio metodologías que me están facilitando en gran medida su desarrollo, depurando mucho más rápido y optimizando los recursos mediante la reducción de polígonos en uso para reducir los tiempos de carga.

5. Referencias

DEV. (s.f.). desarrolladores españoles de videojuegos. <http://www.dev.org.es/>

<https://knowledge.autodesk.com/support/maya/learn?sort=score>

Wikipedia. <https://es.wikipedia.org>

<https://www.gamasutra.com/>

<https://www.stratos-ad.com/>

<http://gamedev.tutsplus.com/>

<https://docs.unrealengine.com/>

<https://www.bensound.com>

<https://docs.unrealengine.com/en-US/index.html>

Artículo disponible en:

<https://www.gametopia.es/learning/article/04/2018/32/disenar-hud-videojuego>

ROLLINGS, A. & MORRIS, D. Game Architecture and Design. New Readers, 2004.

SALEN, K. & ZIMMERMAN, E. The Game Design Reader. Massachusetts Institute of Technology, 2006.

STEPHEN MEADOWS, MARK. Pause & Effect: The Art of Interactive Narrative. New readers. 2002.

SCOTT SPENCER, M. "Introduction To Zbrush 4 With Scott Spencer" Gnomon.

TAVAKKOLI ALIREZA. Game Development and simulation with Unreal technology, 2016. Enlace Disponible en Google Books.

FERRO S. LAUREN. Unreal Engine Blueprints Visual Scripting Projects, 2019. Enlace Disponible en Google Books.

Buenas prácticas de accesibilidad en videojuegos, Colección Estudios Serie Dependencia Nº 12014 (2012)

La dificultad como motivación: subgéneros definidos por el reto en el videojuego contemporáneo Mateo Terrasa Torres, Víctor Navarro Remesal (2015)

Newheiser, Mark (9 de marzo de 2009). «Playing Fair: A Look at Competition in Gaming». Strange Horizons. Archivado desde el original el 12 de marzo de 2009.

Dariush Derakhshani, Randi Lorene Munn (2008). *Introducing 3ds Max 2008*. John Wiley and Sons. p. 240. Retrieved 12 July 2011.

Thompson. *Game design course: principles, practice, and techniques*. Retrieved 16 July 2015.

Arzuza, Johan. ¿Qué es el rigging? (2011). www.artzuza.com

A Taxonomy of Game Engines and the Tools that Drive the Industry

Marcus Toftedahl, Henrik Engström. Proceedings of DiGRA 2019

<http://www.digra.org/digital-library/publications/a-taxonomy-of-game-engines-and-the-tools-that-drive-the-industry/>