



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Facultad de Bellas Artes

Vindicator. Diseño, modelado e implementación de escenografía y utilería para el prototipo del primer nivel de un proyecto personal de videojuego.

Trabajo Fin de Grado

Grado en Diseño y Tecnologías Creativas

AUTOR/A: Rivas Cervera, Francisco

Tutor/a: Martí Ferrer, Francisco

CURSO ACADÉMICO: 2021/2022

TFG

VINDICATOR.

**DISEÑO, MODELADO E IMPLEMENTACIÓN DE ESCENOGRAFÍA Y
UTILERÍA PARA EL PROTOTIPO DEL PRIMER NIVEL DE UN PROYECTO
PERSONAL DE VIDEOJUEGO.**

Presentado por Francisco Rivas Cervera

Tutor: Francisco Martí Ferrer

**Facultat de Belles Arts de Sant Carles
Grado en Diseño y Tecnologías Creativas
Curso 2021-2022**



**UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA**



**UNIVERSITAT POLITÈCNICA DE VALÈNCIA
FACULTAT DE BELLES ARTS DE SANT CARLES**

RESUMEN

En el presente trabajo de tipología práctica se pretende elaborar la creación de entornos e implementación en un motor de videojuego, alcanzando así, estándares de calidad cercanos a la industria. El objetivo es obtener un producto que refleje el conocimiento obtenido durante el grado para completar el perfil de *Environment artist* para juegos AAA. Se logrará a partir de la creación de utilería en 3D, texturizado, nociones sobre el motor de videojuego utilizado, Unreal Engine 5 y demás campos del desarrollo de videojuego.

El título del videojuego es Vindicator pues hace referencia al lore creado para el proyecto, así como contextualización de la época y temática la cual se construye el mundo y componen los distintos contenidos audiovisuales.

Palabras Clave: Diseño escenarios 3D, Videojuegos, Escenografía, Zombies.

ABSTRACT

In this practical typology work, it is intended to elaborate the creation of environments and implementation in a video game engine, reaching quality standards close to the industry. The main goal is to obtain a product that shows the knowledge it needs to become Environment Artist for AAA games. It will be achieved from the creation of 3D props, texturing, notions about Unreal Engine 5 and other fields of video game development.

Keywords: Environment art, Video Games, landscape, Zombies.

CONTRATO DE ORIGINALIDAD

Este Trabajo Fin de Grado ha sido realizado íntegramente por el alumno Francisco Rivas Cervera. Este es el último trámite para la obtención del título de la promoción 2018/2022 del Grado en Diseño y Tecnologías Creativas de la Universidad Politécnica de Valencia. El presente documento es original y no ha sido entregado como otro trabajo académico previo, y todo el material tomado de otras fuentes ha sido citado correctamente.

FIRMA:

FECHA: 22/06/2022

A handwritten signature in black ink, appearing to read 'F. Rivas Cervera', with a large, stylized flourish extending from the bottom right.

ÍNDICE

1. INTRODUCCIÓN

2. OBJETIVOS Y METODOLOGÍA

3. DESARROLLO

3.1 PREPRODUCCIÓN

3.2 PRODUCCIÓN

3.2.1 CREACIÓN DE LOS ESCENARIOS

3.3.1.1 BLOCK OUT

3.3.1.2 MODELADO

3.3.1.3 UV

3.3.1.4 TEXTURIZADO

3.2.2 PERSONAJE

3.3.2.1 MODELADO

3.3.2.2 CREACIÓN ROPA

3.3.2.3 TEXTURIZADO

3.3.2.4 METAHUMAN

3.3.2.5 PERSONAJES SECUNDARIOS: ZOMBIES

3.3.2.6 ANIMACIONES

3.2.3 MOTOR VIDEOJUEGO: UNREAL ENGINE

3.3.4.1 INTRODUCCIÓN

3.3.4.2 ILUMINACIÓN

3.3.4.3 PROGRAMACIÓN IA

3.3.4.4 OPTIMIZACIÓN

3.3.4.5 SONIDO Y PHYSICAL MATERIALS

3.2.4 INTERFAZ DE USUARIO

3.2.5 GUIÓN

3.2.6 RENDERS

4. CONCLUSIONES

5. BIBLIOGRAFÍA

6. ANEXOS

7.1 MEMORIA EXTENDIDA

7.2 ENLACE VIDEOS

7.3 LIBRO DE ARTE

1. INTRODUCCIÓN

Este trabajo parte de mi interés por la escenografía en videojuegos. Motivado por anteriores proyectos personales donde experimenté dicho ámbito, decidí, tras el aprendizaje obtenido, dedicar mi Trabajo de fin de Grado a abordar un proyecto de mayor envergadura, consistente en la elaboración de un escenario para videojuegos, la creación de un personaje y programación y demás elementos de inmersión para lograr una maqueta de juego AAA.

El proyecto nace del interés principal de crear un producto que atienda a los requerimientos necesarios de la industria para formar parte en esta. A su vez, es motivado por otros títulos actuales, series, temática, época representada y demás referentes.

El proyecto pretende ser una experiencia jugable desde la que mostrar distintas técnicas y creaciones en los diferentes aspectos que engloban la creación de videojuegos. A su vez, consolidar conocimientos y maestrías de cara al ámbito profesional.

Se concretó con la creación de una maqueta formada por tres escenarios principales y sus conexiones, personaje principal con sistema de movimiento, enemigos mediante IA, menú principal, diseño de sonido y creación de piezas originales como el tema principal.

El proceso dio paso con la documentación desde un nivel técnico para evaluar la viabilidad del proyecto; búsqueda de una época y contexto (La Inglaterra de 1917 durante un apocalipsis zombie), la construcción de un lore, modelado, texturizado, proyección, programación...

Debido a la magnitud del proyecto, se adjuntan a la memoria una serie de anexos detallando los procesos y detalles realizados, así como la realización de material ilustrativo del proceso y resultados como libro de arte, video renders, video gameplay etc.

2. OBJETIVOS Y METODOLOGÍA

El objetivo principal de este trabajo es la creación de un escenario 3D explorable en tercera persona con las características técnicas y visuales apropiadas para un videojuego AAA, destinado a las recientes generaciones de consolas y pcs con configuraciones *gamer*, ubicado en la Inglaterra victoriana a finales de la segunda revolución industrial¹. Asumiendo la visualización del avatar, se busca una experiencia inmersiva basada en la atmósfera, la iluminación, el sonido, y la posición relativa de la cámara en función de las diversas acciones, con un buen nivel de detalle y profusión de elementos naturales en el entorno.

Para alcanzarlo, se utilizarán técnicas recientes para optimizar el rendimiento gráfico de estas plataformas y se desarrollará un estilo gráfico personal dentro de los parámetros mencionados. El proyecto permitirá aplicar y ampliar las competencias adquiridas durante el grado de Diseño y tecnologías creativas y obtener un producto adecuado para mostrar en mi portfolio competencias propias de un perfil profesional de *environment artist*.

Una vez definidos los objetivos, se procedió con la fase de preproducción, comenzando por barajar qué ideas realizar, sus fortalezas y debilidades. Tras la definición de la idea, se realizó una tarea de documentación sobre la época y lugar y el género al que pertenecería el juego, tomando referentes tanto de videojuegos como de otros formatos narrativos y/o visuales, ya sean pintores cercanos a la época como John Atkinson Grimshaw, series, películas etc, una sinopsis de la historia principal y un resumen de las características del juego abordadas en todas las fases

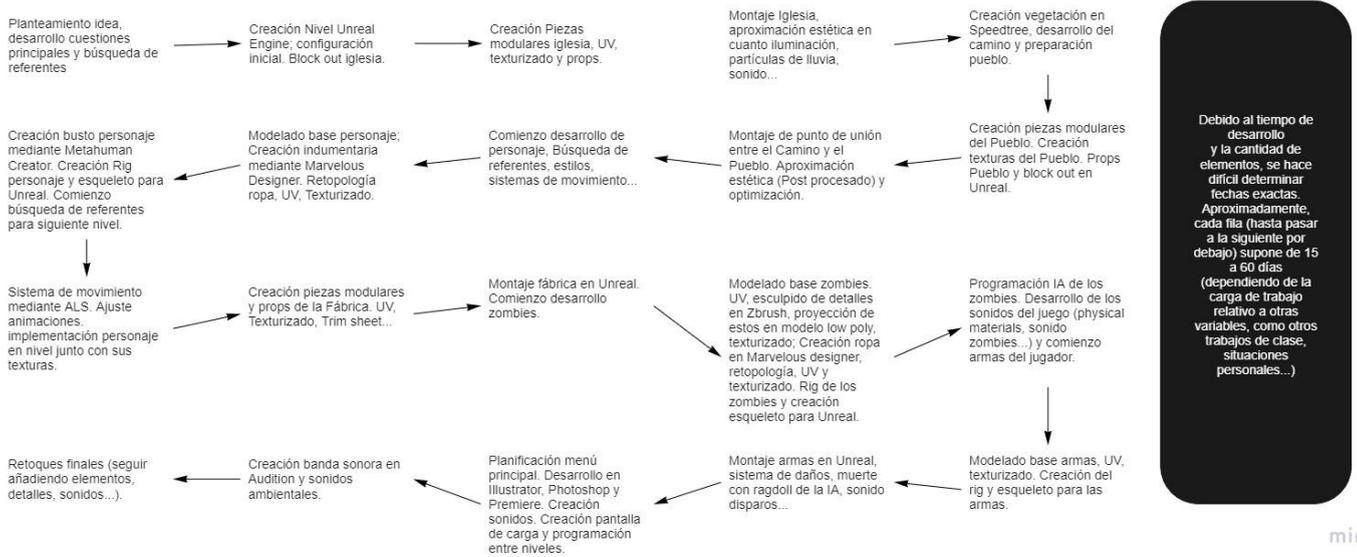
En lo que respecta a la creación de los escenarios, se comenzó creando el nivel en Unreal y realizando las configuraciones iniciales. Se dio paso con una primera fase de *block out*, para elaborar un conjunto de *placeholders* que aproximan la volumetría definitiva en función de las dimensiones del avatar, que serán reemplazados posteriormente por los elementos arquitectónicos y

¹ <https://www.britannica.com/event/Industrial-Revolution>

props definitivos, lo que permitió navegar la escena teniendo una visión de conjunto a medida que estos últimos se van incorporando.

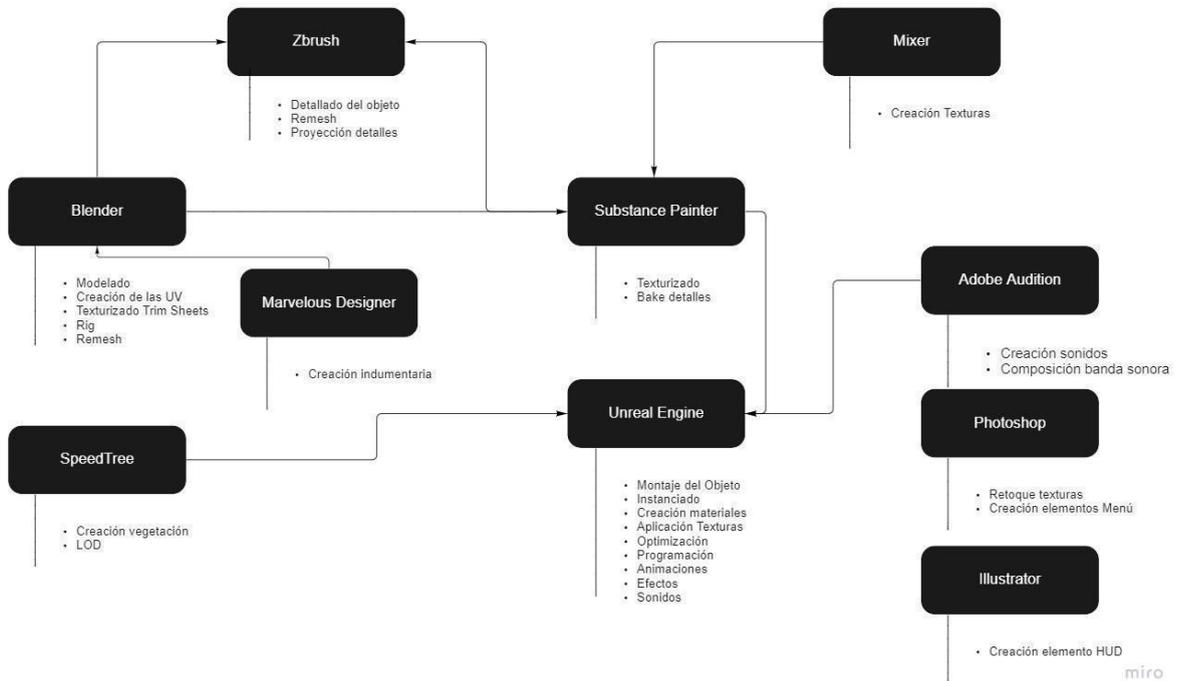
Con esto, se empezó con el proceso de obtención de las piezas modulares de la Iglesia y sus props, UVs y texturizado de los mismos. Al término, se montaron las piezas sustituyendo el block out, se hizo una aproximación de la iluminación y se abordaron efectos como la lluvia. De forma paralela, se trabaja en crear la vegetación en Speedtree y desarrollar el camino que conectaría la iglesia con el pueblo. Tras esto, se comenzó en la elaboración de las piezas modulares que construirían el pueblo, así como los puntos de unión entre el camino y este. Destacar que ya se comenzaba a ajustar la estética de la atmósfera e iluminación mediante postprocesado a la vez que se empezaba a profundizar en cuanto a optimización. Con gran parte del pueblo montado, se comenzó el desarrollo del personaje: modelado, creación de la indumentaria en Marvelous designer, retopología, UV, texturizado y creación del busto mediante el programa Metahuman Creator; creación del esqueleto y riggeado e importación a Unreal para aplicar el sistema de locomoción del personaje. Con esto implementado y con una navegación por el mapa desde un punto de vista en tercera persona, se procedió a cerrar el escenario con la creación de las piezas para la fábrica, así como su punto de conexión con el pueblo. Una vez montado y con todas las partes del mapa ya implementadas (a falta de añadir detalles), se comenzó en el desarrollo de los zombies de la misma forma que el personaje principal, a excepción del busto, que fue modelado en Zbrush. Tras elaborar el esqueleto para los zombies y aplicarse a la malla, se programó la IA y paralelamente, se desarrollaron cuestiones como el sonido, armas y su disparo, sistema de daño para el zombie y planificar el menú principal. Este se llevó a cabo principalmente en Illustrator y Photoshop para los elementos en 2D, Adobe Audition para los sonidos (tanto foley como banda sonora) y premiere para la edición del video y la pantalla de carga. Con todo implementado, conectado y funcionando, se realizaron los retoques finales en cuanto a iluminación, renders para el libro de arte, optimización y *Build* del proyecto para ejecutarse en el equipo encargado de grabar los gameplays.

FASES DEL TRABAJO



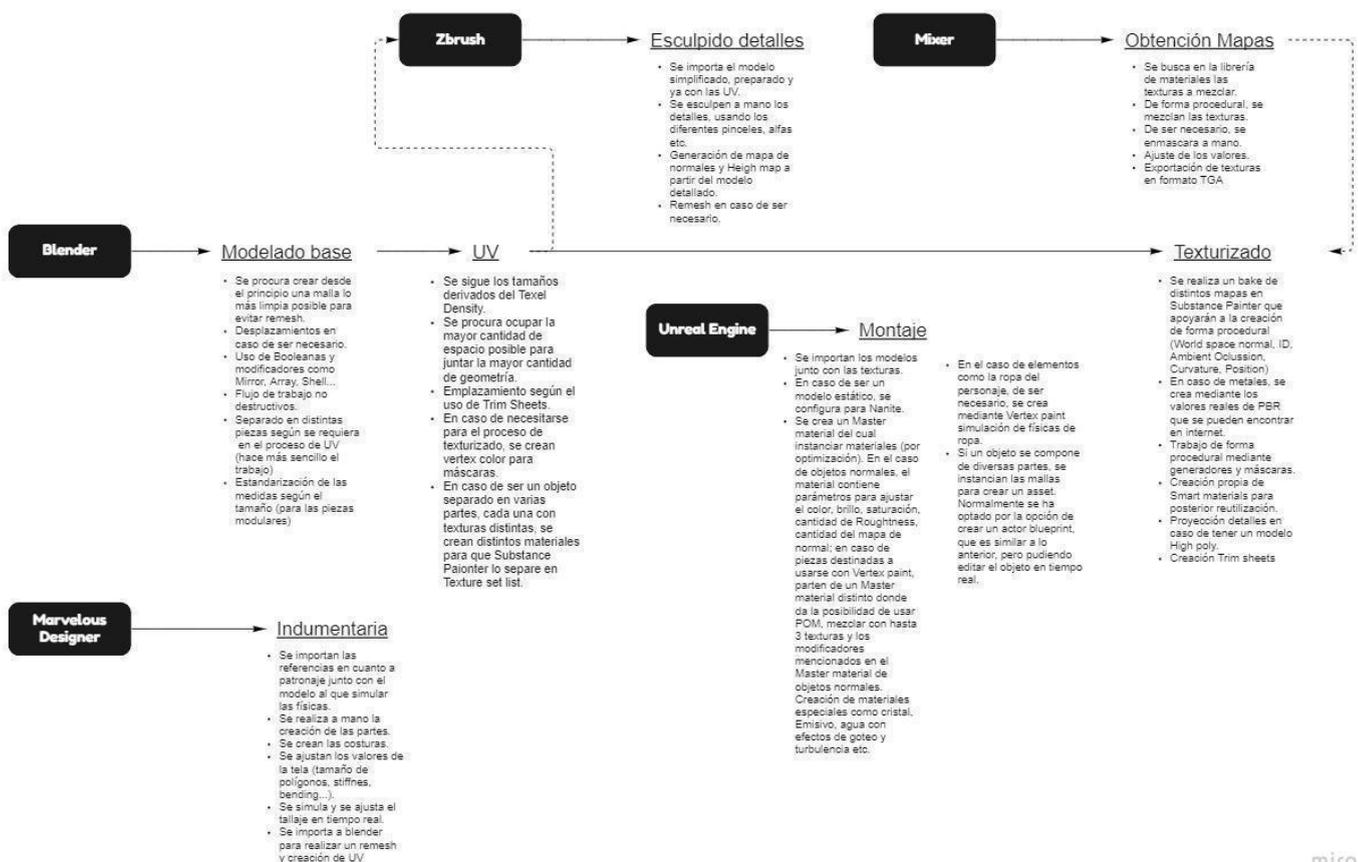
miro

SOFTWARE UTILIZADO



miro

WORKFLOW



miro

Para mayor detalle y acceder a la transcripción, consultar la **Memoria ampliada** en el **Anexo I**.

Importante resaltar la creación de material complementario como la **Memoria ampliada**, en el anexo I, para más información técnica y precisa de lo que se aborde en adelante así como un catálogo de imágenes de lo aquí expuesto y el libro de arte en el anexo II.

3. DESARROLLO

3.1 PREPRODUCCIÓN



El trabajo dio comienzo con su fase de ideación, motivada por títulos de videojuegos AAA tales como *Days Gone*, *The Last of Us* y *Red Dead Redemption 2*, entre otros.

Siendo la intención crear espacios coherentes y creíbles para el jugador, donde los diversos elementos quedasen en consonancia con los aspectos estéticos y la narrativa de un futuro videojuego basado en el presente trabajo fue primordial definir una serie de aspectos para obtener un resultado singular: ¿Qué estética va a tener?, ¿Dónde y cuándo se va a ambientar?, ¿Será un juego de mundo abierto, lineal, por niveles...?



Los primeros pasos del proyecto se dieron en la búsqueda de una localización y época. Motivado por series actuales como *Peaky Blinders*, se decidió optar por ambientar el juego en la época de la segunda revolución industrial de la Inglaterra de principios del siglo XX (Encyclopaedia Britannica, 2021). En este contexto histórico, se optó por plantear una historia con características del género fantástico, incorporando una enfermedad desconocida que transforma a los ciudadanos de las clases trabajadoras en zombies, producida por expansión acelerada de la industria y la insalubridad provocada por el modelo económico de obtención y transformación de los recursos. Esto daría paso a la enfermedad comentada anteriormente que transformaría a los huéspedes en una suerte de zombies.



John Atkinson Grimshaw: *Woman on a path by a cottage*, 1862.

John Constable: *Moonlight Landscape with Hadleigh Church*, 1796.

A Wet Road By Moonlight.

Tras una primera fase de investigación en cuanto a referentes, se dio con el pintor Inglés John Atkinson Grimshaw (*Grimshaw, Atkinson, s.f.*), que realizó durante la época Victoriana (*Wikipedia, La Enciclopedia Libre, 2012*) un conjunto de pinturas que plasman la atmósfera que inicialmente se buscaba y sirvió de inspiración para el trabajo.

En cuanto a la creación de escenarios, las opciones que se plantearon fueron la creación de una iglesia, que funcionaría como baluarte, una ciudad, que constituiría la mayor parte del entorno y una fábrica, que daría acceso a los muelles de la ciudad.

Tras brevemente definir cómo sería el mapa, se comenzó con la búsqueda de referentes en cuanto a entorno, atendiendo cuestiones de estética, época y estereotipos. Se perseguía una estética Victoriana, reminiscencias al *Dark country*, entornos post apocalípticos...



Broady Blackwell: Ghost Story Project, 2020.



Piotr Bystry: Burned old town, 2020.



Peaky Blinders: Fragmento sacado de la serie, 2013-2022.



Peaky Blinders: Fragmento sacado de la serie, 2013-2022.



Eliz Roxs: Thief, 2019.

Fable III: Concept art.

Assassins Creed syndicate: Concept art, 2015

do en el desarrollo de la idea, definiendo al personaje y lo que se iba a construir.

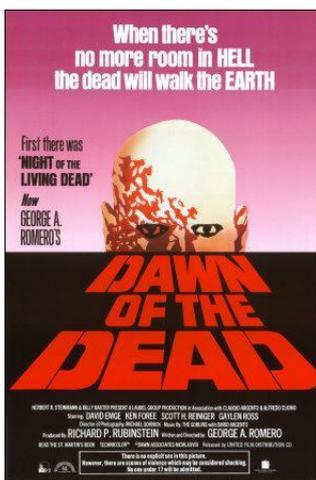


Clint Eastwood: The good, the bad and the ugly, 1968.

Se esbozaron una primera idea de cómo iba a ser el personaje, donde se pretendía seguir el estereotipo y actitudes ya escritas para personajes representados por el actor Clint Eastwood en películas como *The good, the bad and the ugly*, *El sargento de hierro* etc. La psicología del personaje también bebe de series como *Peaky blinders*, de donde se han rescatado referencias visuales del actor Tom Hardy en el papel de Alfie Solomons. Se esbozaron cuestiones relacionadas con el gameplay: sería un juego en tercera persona, que capacidades tendría el personaje, las armas que llevaría etc.



Tom Hardy: Fragmentos de Peaky Blinders en el papel de Alfie Solomons, 2012-2022.



Por último, se buscaron referentes en cuanto al género que se abordaba, Zombies. Se empezaron a buscar precedentes sobre cómo serían, su movimiento, si presentaban deformidades, búsqueda de información sobre enfermedades reales que pudieran respaldar un diagnóstico realista de un virus. Se comenzó con un precedente en el cine de temática zombie, George A. Romero. (*George A. Romero, 2022*)

También se tomaron referentes en otras películas como *Guerra mundial Z*, *Soy leyenda*, series como *The walking dead* y videojuegos como *The last of us* y *Call of duty: Black ops* (Modo zombies) entre otros.

Dawn of the dead: Película dirigida por George A. Romero, 1978.

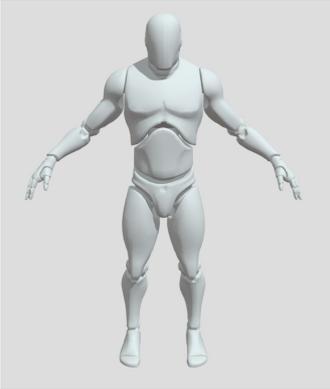
En lo que respecta al software, se eligió Unreal Engine 5, disponible en el momento de comenzar este proyecto en forma de versión *beta* del nuevo motor, debido a las características tecnológicas y visuales que se explicarán más adelante (apartado **Unreal Engine**) que suponían un salto cualitativo importante respecto a otros motores y versiones anteriores del propio Unreal. Aproximadamente en las mismas fechas, la misma compañía (Epic Games) lanzó Metahumans Creator, que se utilizó para crear el busto del personaje. Tras experimentar con estas nuevas aplicaciones y verificar que funcionaban adecuadamente, se apostó por utilizarlas estas nuevas tecnologías para incorporar en el proyecto parte de las posibilidades que ofrecían, buscando resultados profesionales de última generación. Esto supuso diseñar e implementar workflows actualizados para incorporar las nuevas herramientas y procesos disponibles.

Epic Games: Fragmentos de Lumen in the land of nanite, presentación de Unreal Engine 5, 2021.



3.2 PRODUCCIÓN: CREACIÓN ESCENARIOS

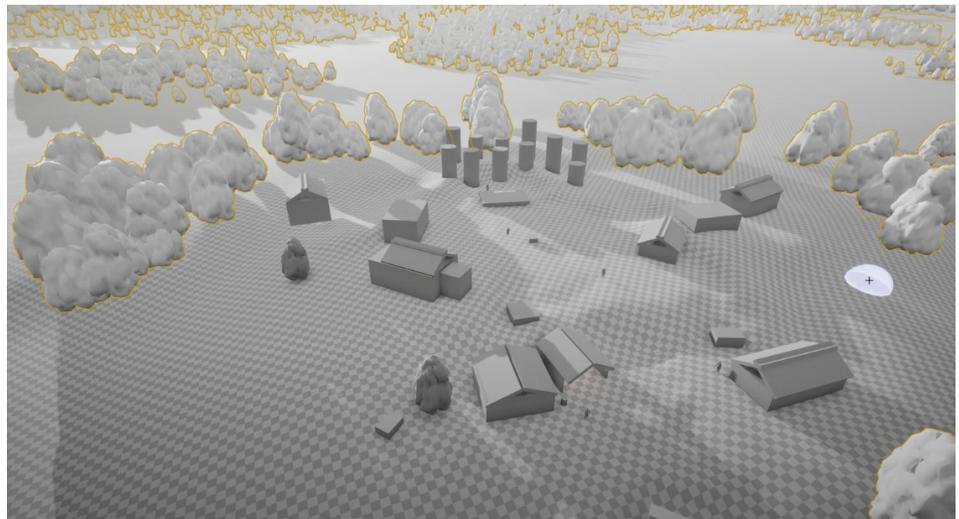
BLOCK OUT



Mannequin de Unreal Engine 4 en su pose predefinida

Ejemplo de proceso de block out

El block out o placeholder es la parte inicial en el desarrollo de los escenarios donde, valiéndose de figuras geométricas simples, se construye una primera maqueta donde clarificar cuestiones como proporciones, espacios, luz (saber desde dónde viene)... Para asentar las distancias y tamaños, se ha usado el *mannequin* por defecto de Unreal Engine, con una estatura de 183 cm.



Para la creación del terreno en Vindicator se han utilizado las herramientas de modelado que ofrece Unreal.

Posteriormente, se creó un material específico que permitía la mezcla de 5 texturas distintas, utilizado para crear distintos tipos de terreno.

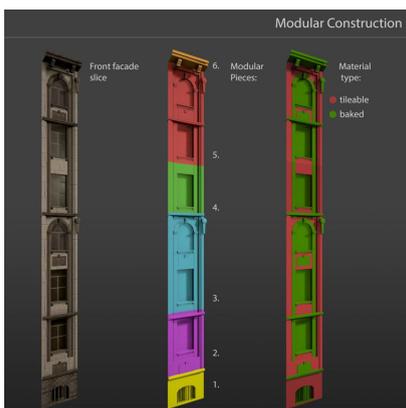
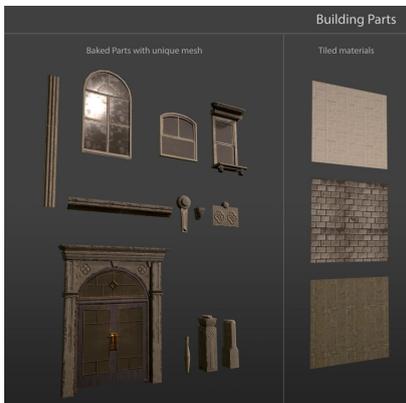
Para generar y posicionar la vegetación, en el mismo material creado, se asignaron distintos tipos de vegetación, su densidad y distribución dependiendo la textura que se pintara. De esta manera se podían crear distintos tipos de biomas de forma procedural, pudiendo en caso de necesitarse, ajustar parámetros, añadir y/o quitar vegetación.

MODELADO

A un nivel general, podemos dividir los modelos 3D en dos tipologías: los elementos arquitectónicos y la utilería (*props*).

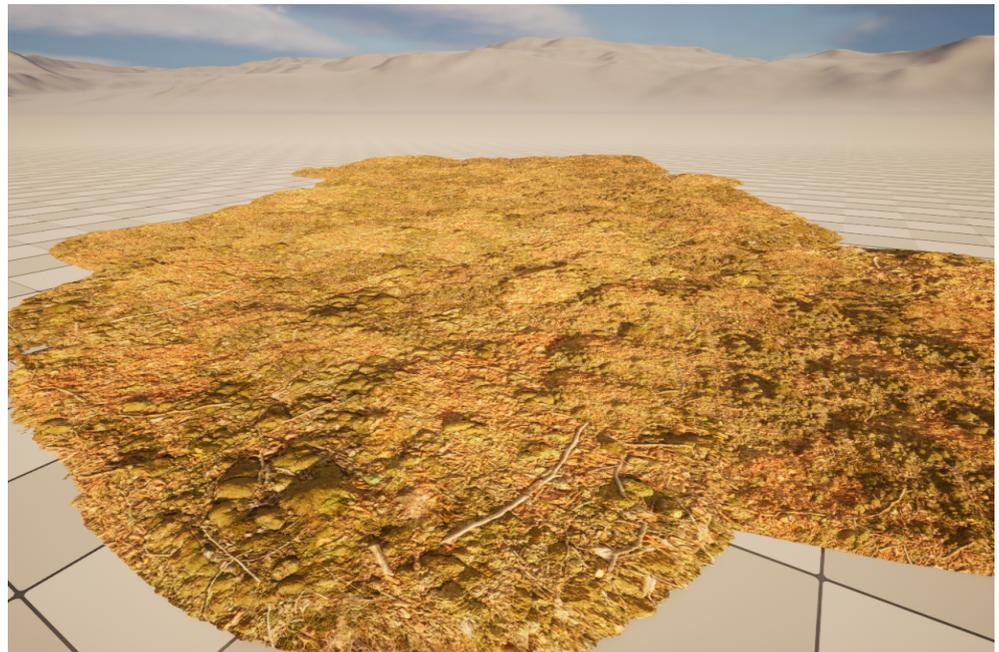
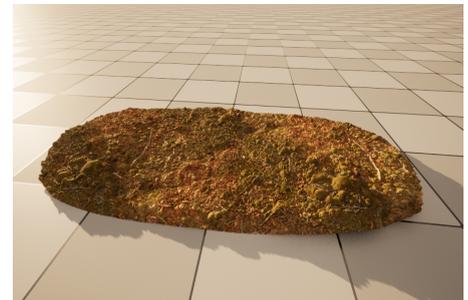
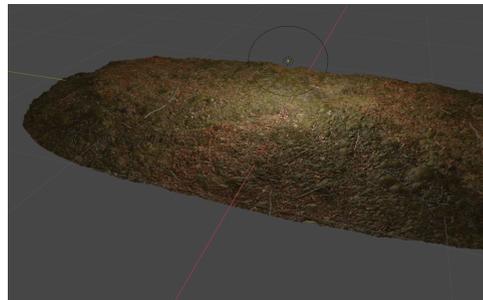
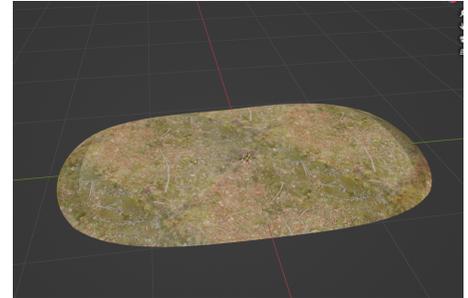
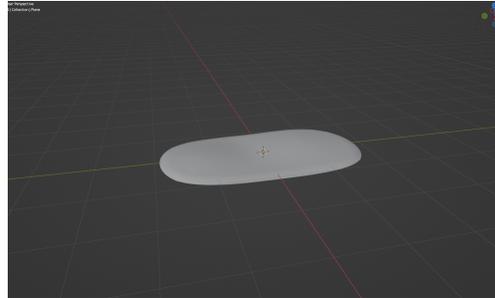
Para la construcción de los edificios, se ha seguido el método de construcción modular. Este consiste en crear diferentes piezas con dimensiones estandarizadas para poder ser montadas conjuntamente de manera variable encajando entre ellas. Esto implica dividir la arquitectura en bloques repetibles (un fragmento de muro con una ventana insertada, un pilar, una pared simple, etc). Esta técnica permite crear rápidamente construcciones complejas de forma óptima, ahorra tiempo a nivel de texturizado, pues comparten texturas, lo cual ayuda a su vez a la optimización del juego, dado que el instanciado de objetos permite almacenar cada objeto repetido una única vez. Para proporcionar variedad suplementaria, esta forma de trabajar se complementa con el uso de diferentes texturizados y con la creación de ornamentación y *props*.

También, la aparición de *Nanite* (Descubre La Importancia de Nanite, 2021), que permite la utilización de geometrías con una gran cuenta de polígonos sin sacrificar el rendimiento en tiempo real, se ha aprovechado para poder crear zonas detalladas del terreno modeladas manualmente con una gran definición. Para poder crear estos *assets*, la técnica adoptada ha sido utilizar el mapa de desplazamiento o *heightmap*, a partir de la subdivisión de una malla base.

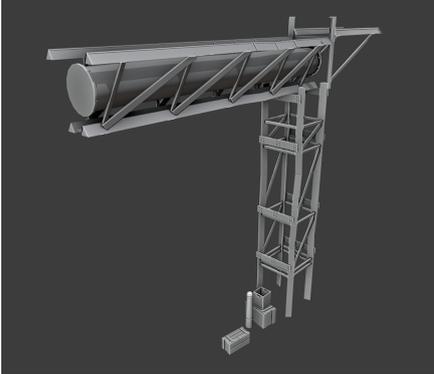


Modular Buildings: Tips and Tricks,
2019

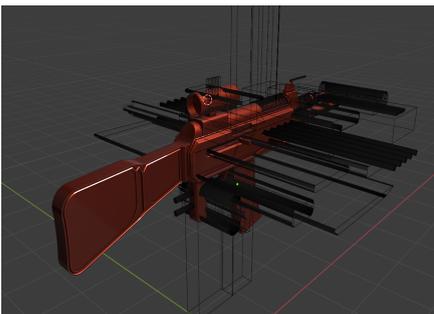
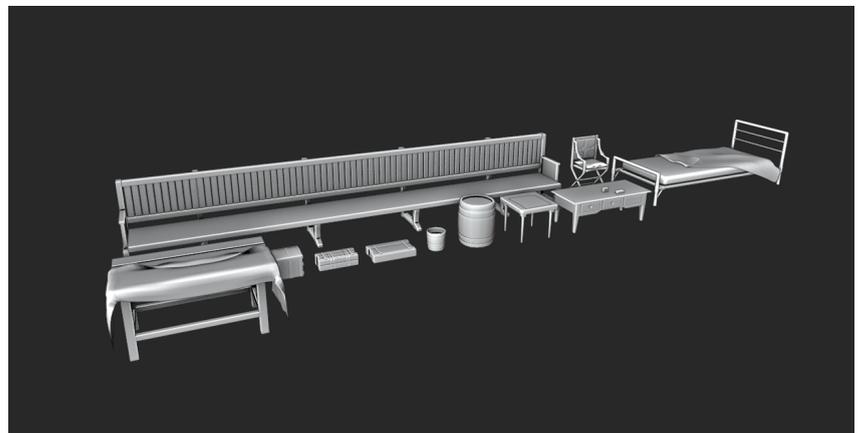
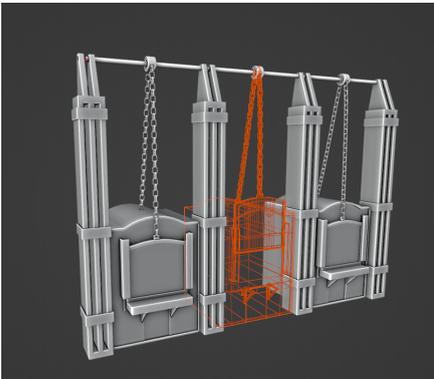
Ejemplo de creación de prop a partir del proceso de desplazamiento para Vindicator y como se muestra dentro de Unreal Engine 5



A pesar de la libertad de poder utilizar una enorme cantidad de polígonos, todos estos assets han sido decimados para optimizar el rendimiento.



En cuanto al modelado de mobiliario, maquinaria, instalaciones diversas y utilería, se ha procurado dotar a los objetos de una topología lo más limpia y optimizada posible. En los elementos que lo requerían, se ha utilizado un *workflow* consistente en crear un modelo de alta poligonización para posteriormente, realizar una retopología y proyectar los detalles mediante un mapa de normales. Sin embargo, en la mayoría de los casos se ha construido la topología directamente en la fase de modelado con una poligonización baja o media en función de cada objeto particular sobre el cual, añadir los detalles en el *software* de texturizado.



Otro de los métodos que se ha usado para la creación de *props* es mediante el uso de operaciones como booleanas, caracterizado por no ser destructivos. Esto permite realizar hendiduras, huecos, formas sin deformar la malla y pudiendo editarlo en cualquier momento. Posteriormente, se ha de limpiar la malla para evitar problemas de *shading* o *Ngons*. Para realizar este proceso de forma más sencilla, se han utilizado los plugins de **Box cutter** junto con **Hard ops** (About BoxCutter - BoxDocs, s.f.).

Ejemplo de props modelados para Vindicator.

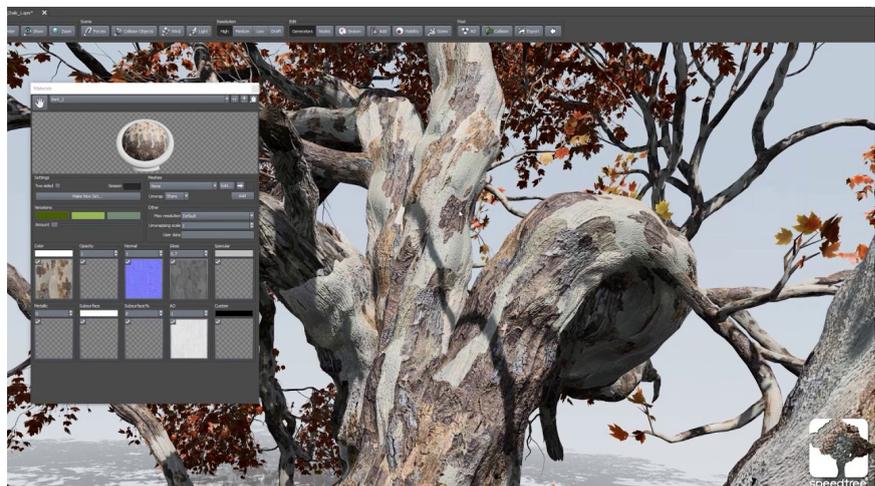
Ejemplo prop desarrollado para un proyecto personal, creado a partir de booleanas y siguiendo un proceso no destructivo.



Malla de las armas creadas para Vindicator y su render con materiales

Para la creación de vegetación, se ha usado el *software Speedtree*, que contiene herramientas de creación destinadas a este tipo de *assets*. Sin embargo, debido a la cantidad de vegetación que había que desarrollar y la falta de tiempo, también se ha optado por utilizar modelos de uso gratuito y libre de derechos de la mano de *Quixel*.

Captura del programa Speedtree utilizado para elaborar vegetación



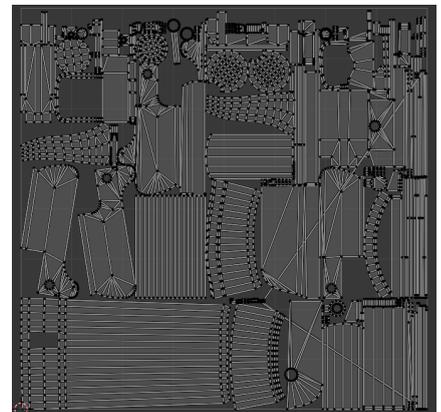
UV

Como ya se ha avanzado anteriormente, mantener un correcto mapeo de las UV del objeto, permite alcanzar mayor resolución sin necesidad de utilizar tamaños de textura elevados. También es importante mantener la misma densidad de texel.

Los modelos son separados en distintas partes para mantener todos la misma densidad de píxeles. A su vez, se intenta ocupar el mayor espacio posible y juntar la mayor cantidad de geometría para reducir el exceso de mapas.



Captura de las costuras de una de las partes del arma para Vindicator

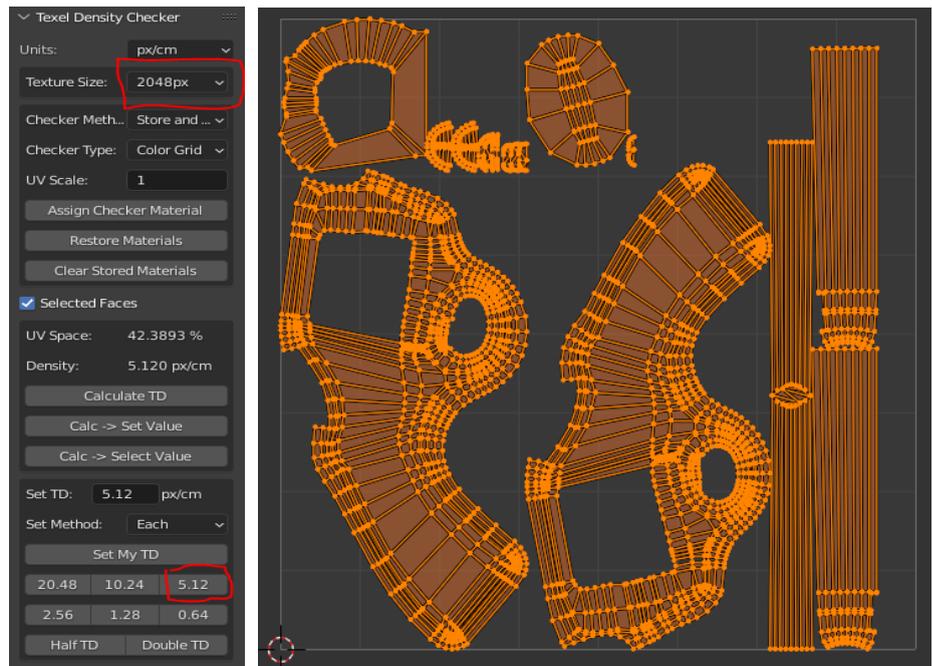


Captura del despiece de UV de la parte del arma mostrada en la figura anterior

En cuanto a técnicas y *plugins* utilizados, para la mayoría de geometrías con curvas, se han creado las costuras a mano, pues es la opción más flexible; los demás objetos se han mapeado mediante la opción de *Smart UV project*, que crea las costuras en base a los ángulos en la malla. El predeterminado es 66°.

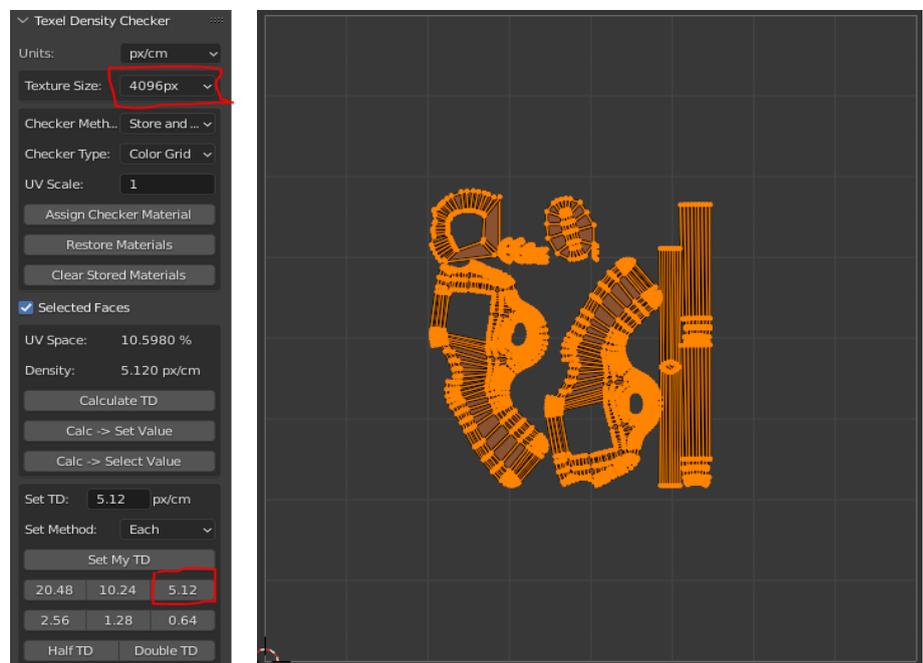
Uno de los plugins utilizados para el proceso de mapeado, **Texel density checker**, que te permite ajustar el tamaño de las UV en base al tamaño de la textura a utilizar y los px/cm - px/m.

Captura del proceso explicado anteriormente sobre el proceso de ajustar el texel density. Se muestra el espacio ocupado en las UV para los parámetros de la figura anterior



En las imágenes se aprecia como, usando un tamaño de texturas de 2048px a un TD de 5.12cm, da como tamaño resultante el de la imagen anterior. Si por el contrario, se utilizase un tamaño de textura de 4096 manteniendo el TD, el resultado es el siguiente:

Captura del proceso explicado anteriormente sobre el proceso de ajustar el texel density. Se muestra el espacio ocupado en las UV para los parámetros de la figura anterior

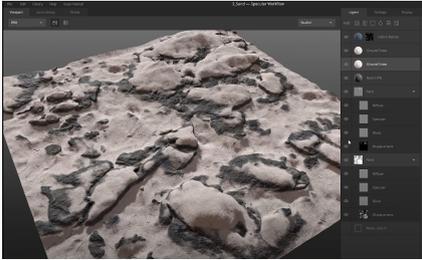


Sobre el por qué utilizar un TD de 512px/m, se ha seguido la tabla elaborada por el artista *Leonardo Lezzi* (Leonardo lezzi - Lead Level Artist, s.f.).

128	PX/M =	1.28	PX/CM	STRATEGIC CAMERA
256	PX/M =	2.56	PX/CM	STRATEGIC CAMERA
512	PX/M =	5.12	PX/CM	THIRDPERSON CAMERA
1024	PX/M =	10.24	PX/CM	FIRST PERSON CAMERA
2048	PX/M =	20.48	PX/CM	FINAL RENDER
4096	PX/M =	40.96	PX/CM	HERO SHOTS

Leonardo Lezzi: Tabla de resolución
en base al punto de vista para texel
density

TEXTURIZADO



Captura del proceso de creación de texturas desde el software Mixer

Para poder crear las texturas necesarias y que se adapten a las necesidades del proyecto, se ha recurrido a la herramienta *Mixer*.

Esta permite acceder a una amplia librería de texturas fotográficas de uso libre, para, posteriormente, aplicar y ajustar paramétricamente diferentes técnicas de mezcla entre texturas, mapas de desplazamiento, pinceles para retoques a mano etc.

La mayor peculiaridad del programa es la capacidad de mezclar texturas enmascarando de forma procedural, obteniendo una gran calidad y una gran variedad.

Finalmente, las texturas son exportadas en formato TGA (targa), debido a la compresión similar a formatos PNG pero con mayor ahorro de espacio.

Una vez creadas las texturas necesarias, se pasa a *Substance Painter* para su texturizado. Este programa permite crear materiales que se ajusten mediante, por ejemplo, mapas de Curvatura (*Curvature - Substance 3D Painter*, n.d.), *Heightmap* etc. También permite el control de estos parámetros, ajustándose a la malla de objetos distintos automatizando el proceso.



Ejemplo de Trim sheets

Sin embargo, esta técnica no es aplicable a todos los objetos, pues a parte de que genera mapas únicos para un solo objeto, lo cual perjudica a nivel de optimización, debido a parámetros ya comentados anteriormente como el *Texel density*, habrá objetos que por su tamaño, requerirán de ser tileados, como rocas, grandes superficies etc.

Otra de las técnicas utilizadas en la industria por su optimización y flexibilidad que ofrece, es el uso de *Trim sheets*; esta técnica permite que distintos objetos compartan la misma textura, teniendo la particularidad de que está compuesta por diferentes diseños y permite la repetición hacia uno de los lados.

La forma en la que se ha elaborado para este proyecto, ha sido a partir de un plano con extrusiones en partes concretas, que ha sido exportado hacia Substance y desde donde se ha creado diversos diseños; También se suele crear en Zbrush, esculpiéndose los detalles necesarios y, posteriormente, haciendo un *bake*.

Una vez se tienen las UV del objeto a su tamaño adecuado, se sitúa en el lugar donde se quiera proyectar el detalle.

Otras formas para mezclar materiales o evitar la repetición, es el uso del vertex paint dentro del motor de videojuego.

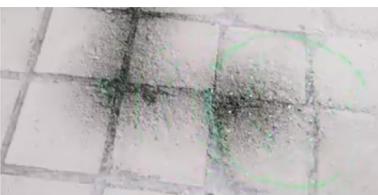
Esto aprovecha los vértices en la malla para aplicar 3 valores (RGB, rojo, verde y azul) a cada polígono y actuando a modo de máscara.

Esto permite aplicar hasta 3 materiales distintos a un objeto.



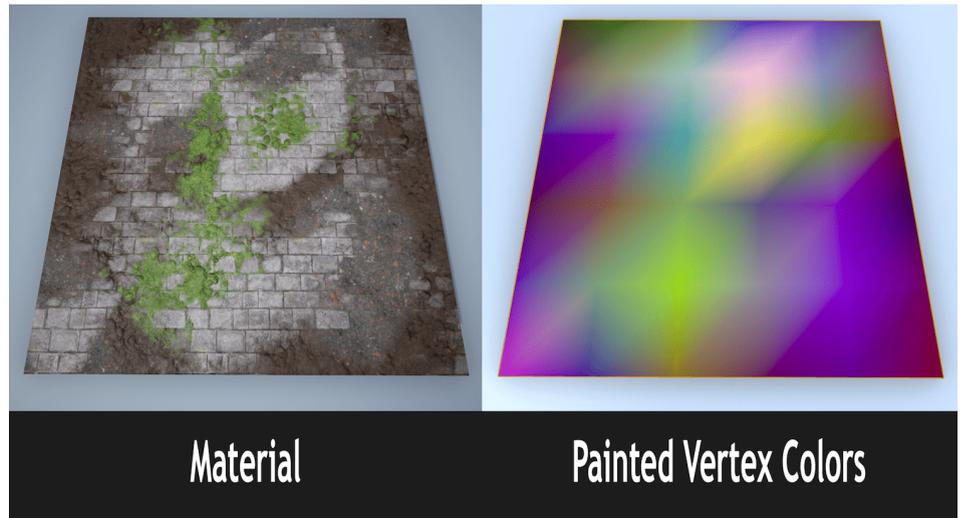
Ejemplo del uso de Vertex paint, visualización con materiales y la mezcla RGB.

Captura de ejemplo sobre el uso de funciones para crear variación en los bordes



Ejemplo del uso de Vertex paint con una mezcla mediante el Heightmap

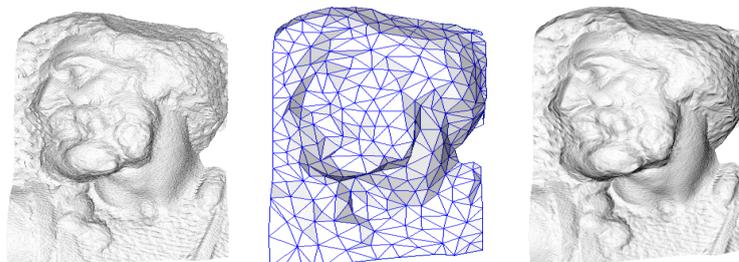
Proceso de bake de detalles desde un modelo de alta poligonización a otro de baja poligonización



Para difuminar la transición entre materiales, se pueden crear funciones para aplicar un mapa de ruido a los bordes.

A su vez, también se puede usar un *Heightmap* para mezclar a partir de este. Sin embargo, esto tiene sus limitaciones, pues Nanite no soporta el vertex paint, con lo que los objetos que vayan a precisar de esta técnica, no van a poder beneficiarse de esto.

Añadir, como se había avanzado anteriormente, la proyección de detalles, que consiste en proyectar mediante un mapa de normales la información de una geometría detallada a otra más simple.



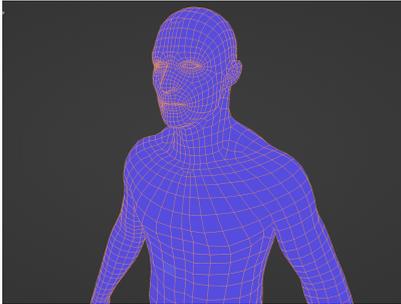
original mesh
4M triangles

simplified mesh
500 triangles

simplified mesh
and normal mapping
500 triangles

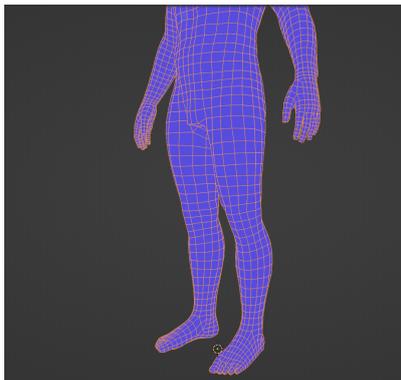
3.2 PRODUCCIÓN: PERSONAJE

MODELADO



Para la creación del personaje, se ha partido del mismo *mannequin* utilizado durante la construcción del escenario para las medidas y proporciones. Se comenzó con una malla base aunque, más tarde al elaborar la indumentaria, se desechó.

El modelo debía de tener unas proporciones correctas y presentar un *flow* correcto de la topología, pues la indumentaria que se elaboraría a continuación, se iba a realizar en Marvelous Designer, programa que utiliza la simulación de físicas para el ajustado en el avatar.

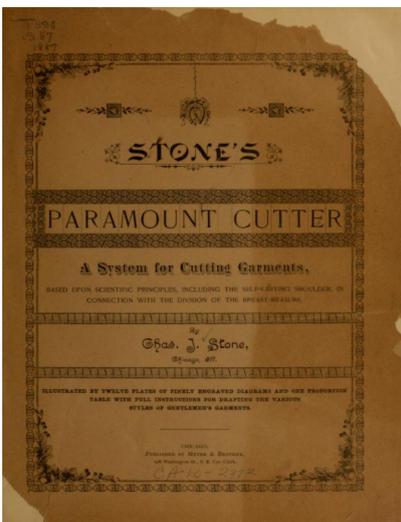


ROPA

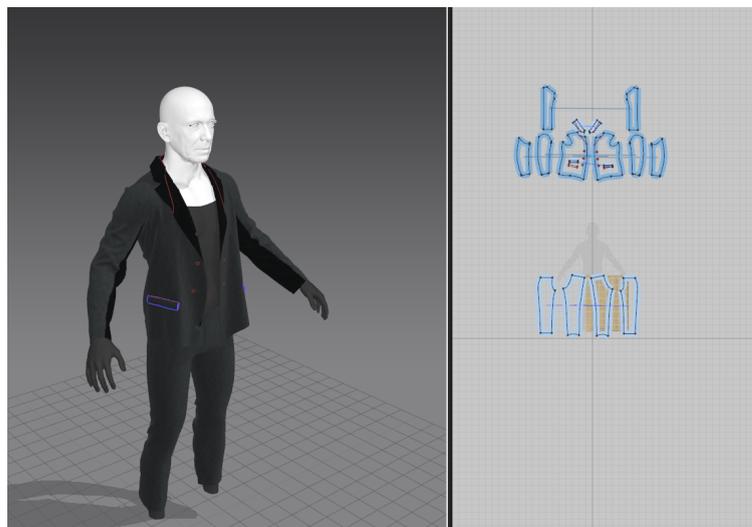
En la realización de la indumentaria del personaje, se realizó mediante el *software* de Marvelous Designer, donde se elabora mediante patrones reales y posteriormente simulando su ajustado al avatar, dando detalles como arrugas, costuras y pesos de gran fidelidad.

En la búsqueda de información, pues se tenía que ajustar a los cánones de indumentaria de la época y el lugar, se dio con un libro publicado en 1887, donde sacar los distintos patrones para el proceso. (Stone & The Library of Congress, 1887)

Capturas de la malla base del protagonista en Vindicator



(Stone & The Library of Congress, 1887)



Capturas del proceso de creación de la indumentaria para Vindicator



Capturas de la malla final de la indumentaria tras el remallado

Posteriormente, se trasladó el modelo a Zbrush para añadir detalles y realizar una retopología de la malla, siendo finalmente importado a blender, donde se crearían los UVs del modelo para ser texturizado.

TEXTURIZADO

El texturizado se llevó a cabo en Adobe Substance 3D painter, antiguo Substance Painter, donde siguiendo con referentes históricos en cuanto a la utilización de materiales (*Early 20th Century Fabric Glossary, 2020*), se procedió al texturizado. A su vez, se le añadieron detalles tales como suciedad, desgaste y mojado, pues debía de adecuarse al entorno, el clima y la estética. Se emplearon técnicas comentadas anteriormente como el uso de mapas de curvatura.



Capturas tras el proceso de texturizado

Se realizó un bake de ciertos mapas en una única textura: Oclusión ambiental, *roughness* y *metal*, se guardan en los distintos canales RGB, de modo que se optimiza la cantidad de mapas (*Documentation – Arm Developer, s.f.*).

METAHUMAN

Para la creación del rostro, se optó por realizarse mediante la aplicación en línea de *Metahuman Creator*. Esta aplicación ha sido creada por Epic Games, de donde pertenece el motor Unreal Engine. Metahuman Creator permite la creación de personajes de alta calidad, ya *riggeados* y con facilidad de integración en Unreal Engine.

Únicamente se necesita el busto, pues el cuerpo será la propia ropa elaborada previamente.



Capturas tras el proceso de creación del busto del personaje mediante el software de Metahuman

PERSONAJES SECUNDARIOS: ZOMBIES

El proceso de creación es muy similar al del personaje principal; se reutiliza la malla base y se modifican algunas partes para adaptarlo a los requerimientos. El modelo es llevado a Marvelous Designer para crear la ropa. Se realiza una retopología, se separa el modelo en partes para mayor tamaño en el mapeado de UVs; El busto es llevado a Zbrush para ser esculpido y, posteriormente, proyectar el mapa de Normales y Desplazamiento en la malla de baja poligonización.

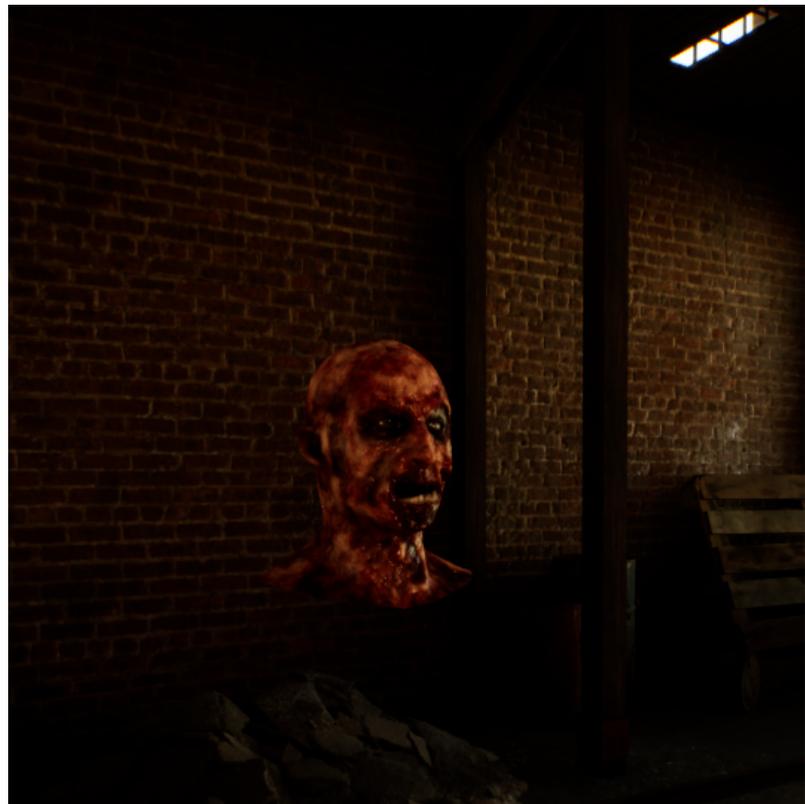
Se importa el modelo en Substance Painter para su texturizado y posteriormente a Unreal para cerciorarse que las texturas estén correctas.



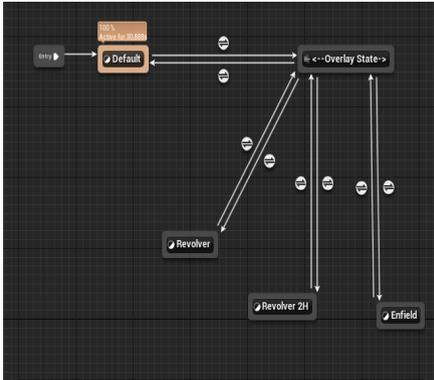
Modelo base de los Zombies.

Captura tras el texturizado del busto de los zombies.

Captura desde el motor de videojuego de como luce el modelo final.



ANIMACIONES



Captura de una parte de la máquina de estados del protagonista en Vindicator. Muestra los estados de Default, Rifle, Pistol 1h y Pistol 2h, para determinar el objeto que lleva en la mano y cambiar las animaciones en consecuencia

Uno de los aspectos fundamentales en el desarrollo de videojuegos es el movimiento del personaje; desde cómo se mueve, las animaciones, dinamismo, velocidad, capacidad de movimientos etc.

Es importante definir previamente qué capacidades de movimiento tiene el personaje. En este proyecto, sabiendo que lo principal era la construcción de escenarios, se amplió creando un pequeño movimiento para mayor inmersión y para situar mejor al espectador sobre cómo se vería el mapa desde la perspectiva en tercera persona.

Normalmente el movimiento es construido mediante árboles de comportamiento, que ejecutan ramas de código que contienen elementos como el propio desplazamiento, animaciones etc.

En este caso, se ha aprovechado un sistema de movimiento como es el *Advance locomotion system* (Advanced Locomotion System v4 in Blueprints - UE Marketplace, s.f.). Este sistema provee de movimientos con animaciones y transiciones entre ellas. También provee de elementos más complejos como *virtual bones*, *IK foot* y un sistema de cámara a medida.

Este sistema de locomoción ha sido ampliado con distintas funciones: Función de disparo (*Spawn* de proyectil, que siga un trazado concreto, detecte colisión al impactar y sobre qué ha impactado; Sonido de disparo etc), mejora de la cámara (añadiendo movimientos orientados principalmente a poder inspeccionar el terreno). Se han añadido *stats* de vida, estamina y escudo (a pesar de ser desechados al final), detección de *Physical materials* para añadir distintos sonidos de pasos y aparición de VFX como polvo o tierra al caminar.



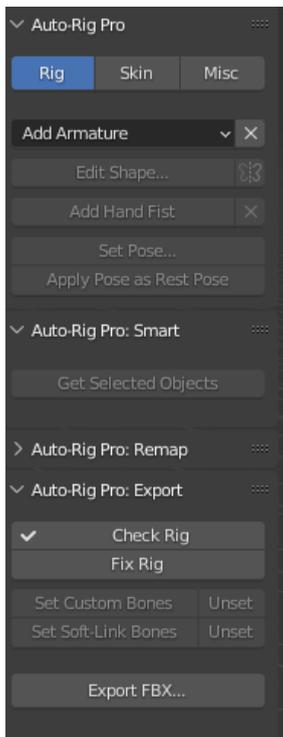
Listados de todos los huesos, virtual bones y sockets utilizados para el esqueleto en Vindicator

Al trabajar mediante un sistema ya creado (debido a que partimos de un esqueleto con unos huesos predefinidos y *virtual bones*), se debía de producir un esqueleto con el mismo nombre de los huesos y su posición o, en su defecto, realizar un *retarget* de todas las animaciones al nuevo esqueleto, añadiendo posteriormente los *virtual bones* y volviendo a configurarse.

Debido a experiencias previas con lo segundo, donde los resultados no eran del todo satisfactorios, se procedió a clonar el esqueleto para evitar realizar un *retargeting*. Puesto que el proceso de hacerlo a mano es muy tedioso, y al tener *virtual bones* tampoco se podía simplemente aplicar el esqueleto a la malla, tras buscar alternativas se dio con un *plugin* para Blender que te genera el esqueleto mediante un sistema similar al de Mixamo, pudiendo al exportar, que se renombre los huesos específicamente para Unreal Engine a la vez que se añaden los *Virtual Bones*. El *plugin* se llama *Auto-rig Pro*.

Una vez terminado creado el esqueleto y aplicado a la malla del personaje, al importar a Unreal, habrá que, o bien cambiar el esqueleto que usa el personaje importado por el de ALS, o lo que se ha realizado, que es sustituir la malla del ALS por el personaje. Finalmente, lo único que queda es reconstruir el cuerpo de físicas para las colisiones, a parte de que el personaje contiene aspectos como un *Ragdoll* o acciones como rodar o simulación de ropa.

Las armas a su vez, cuentan con un esqueleto y sockets propios, desde donde *spawnear* proyectiles etc. Los ajustes realizados y demás información adicional se encuentran en el anexo **Memoria ampliada**.



Captura del plugin Autorigpro, utilizado para crear el esqueleto



Comparación de mallas distintas con un mismo esqueleto

3.2 PRODUCCIÓN: MOTOR VIDEOJUEGO UNREAL ENGINE

INTRODUCCIÓN

Para integrarlo en un motor de videojuego auténtico, donde crear una experiencia jugable y un renderizado en tiempo real, se ha usado Unreal Engine 5.

Los diversos motivos de la elección de este motor es debido a varios factores: Es gratuito, hasta el punto de la venta del producto, donde únicamente pagas *royalties* cuando el producto supere el millón de dólares de ingresos; actualmente, de los motores de acceso gratuito, es el motor más desarrollado y con mayores tecnologías enfocadas a la creación de títulos AAA, siendo tecnologías mencionadas anteriormente como *Nanite* o *Lumen* (este último, el sistema de iluminación) un gran paso en la industria.

Por su familiaridad puesto que anteriormente ya se había realizado proyectos personales y se tenía una noción previa.

Cabe destacar que muchas compañías grandes están optando por este motor, dejando de lado sus propios motores (es común que muchas compañías grandes desarrollen sus propios motores gráficos, lo que supone grandes costes de desarrollo). Ejemplos de esto son CD Projekt, que traslada el desarrollo de su nuevo título de *The Witcher* (Andy Chalk published, 2022) ó Crystal Dynamics, que lleva la saga de *Tomb Raider* (*Un Nuevo Tomb Raider Está En Desarrollo Con Unreal Engine 5 Por Crystal Dynamics, s.f.*).

Esta expansión que ha recibido durante los últimos años ha traído consigo la aparición de una comunidad extensa.

ILUMINACIÓN



Una de las principales cualidades que ha traído la nueva versión de Unreal Engine, ha sido su nuevo sistema de iluminación. Conocido como *Lumen*, es un sistema de iluminación y reflejos totalmente dinámico diseñado para nuevas generaciones de consolas. La piedra angular de este sistema es, a grandes rasgos, la generación de iluminación indirecta propagando el color en las superficies cercanas (cuya visualización dependerá del nivel de brillo de los pixels de la textura utilizada) mediante un sistema de *Surface Cache* (Utiliza *ray tracing* por software). (*Lumen Global Illumination and Reflections*, s.f.)

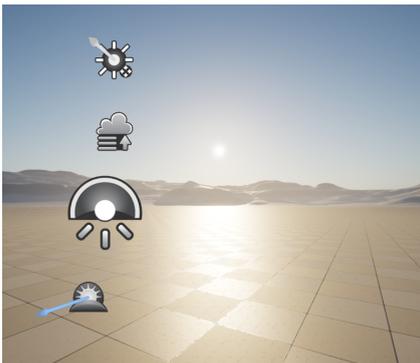
Renders de ejemplo sobre el uso de Lumen, por Epic Games.



Como añadido, debido a que Lumen funciona generando *cartas* del objeto que se almacenan en un atlas de baja resolución (todo esto de forma interna en el motor), es importante dividir los escenarios en piezas (por ejemplo, techo, paredes y suelo de una habitación) para una mejor captura por parte de estas cartas.

Uno de los mayores retos a la hora de plantear Vindicator fue la decisión de tener luces completamente dinámicas. Esto significa no poder *bakear* la iluminación y así poder aliviar la carga de renderizado, lo que llevaría a, siendo la iluminación uno de los aspectos que más consume, tener cuidado con la praxis a la hora de iluminar el escenario.

Para consultar cómo se ha construido la iluminación así como los efectos y postprocesado, consultar ***Memoria ampliada***.



Ejemplo sobre el cambio de iluminación debido a como interactúa el brillo del mapa de Color junto con la iluminación indirecta.

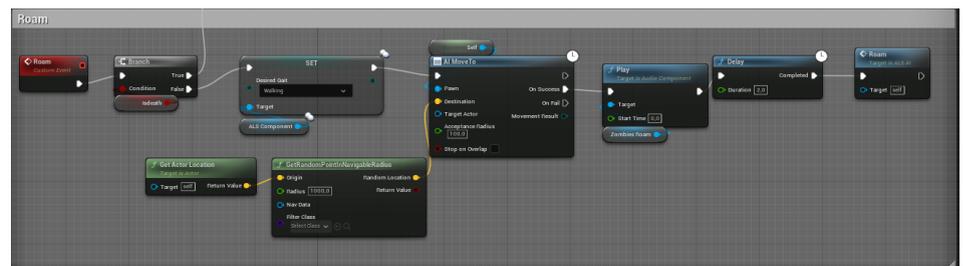
Captura de todos los actores que intervienen en el proceso de iluminación.

PROGRAMACIÓN IA

En la creación de la IA para el proyecto, se barajaron diferentes opciones. Una de ellas era elaborar una máquina de estados de personaje, pero orientada a los comportamientos de la IA. Sin embargo, y debido a que no se necesitaba para las motivaciones del proyecto una IA compleja, se decidió simplemente programar diferentes estados.

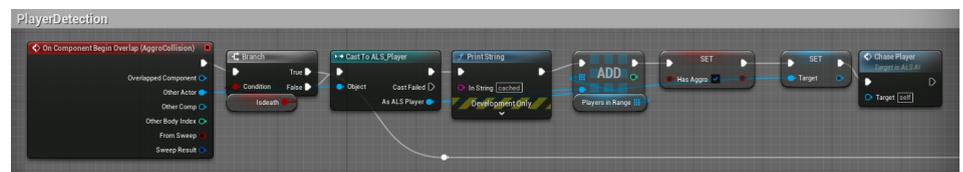
El primero de ellos, denominado como *Roam*, se basa en generar un punto de forma aleatoria a partir de la posición del zombie en un radio predeterminado. El zombie irá a ese punto, donde se parará por 2 segundos hasta volver a generar otro punto e ir hacia el. Esta acción solo se ejecuta si el zombie sigue vivo; a su vez, reproduce un sonido. Destacar, que el movimiento del zombie al realizar esta acción será andando.

Línea de código del estado Roam.



El siguiente estado, es la forma de detectar al jugador. Cada zombie tiene una esfera invisible con un tipo de colisión creado específicamente. Cuando el jugador (que también posee este colisionador) entra en el radio del zombie, es llevado a un array donde se guarda e inicia el siguiente estado o evento: *Chase player*.

Línea de código destinado a la detección de personajes.



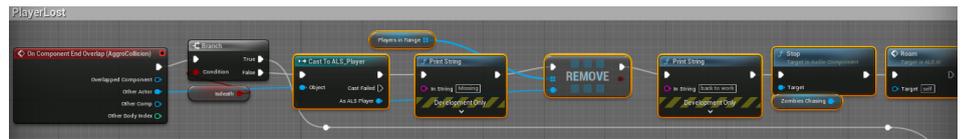
Este estado se basa en el mismo principio que el de *Roam*, la diferencia es que en vez de generar un punto al que ir, el zombie se mueve hacia el punto del jugador, actualizándose cuando este se mueva de manera que pueda simular una persecución. El zombie pasa al estado de correr, moviéndose a más velocidad, y el sonido que emite son gritos característicos. Destacar que esta acción nunca llega a ser cumplida, de forma que la única manera de que el zombie deje de perseguir al jugador es, o bien acabando con él, o saliendo del radio de colisión.

Línea de código destinado a iniciar el estado para perseguir al jugador.



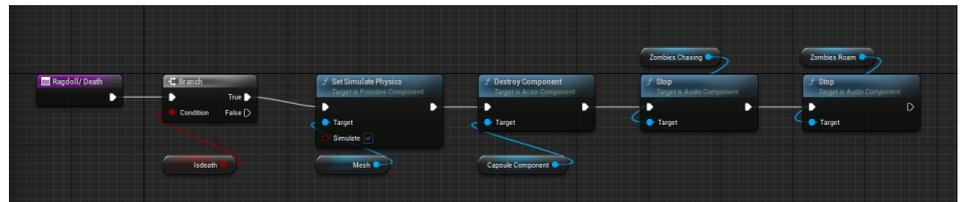
Salir del radio de detección activa el siguiente evento, *Player Lost*, donde se elimina del array al jugador e inicia el evento de *Roam*.

Línea de código para cuando el jugador sale fuera del rango de persecución.



Para la muerte, el zombie detecta en qué parte impacta la bala (recoge el hueso del esqueleto donde golpea), donde se crea un pequeño impulso para dar *feedback* al jugador. En caso de recibir dos impactos de bala, el zombie simulará un *Ragdoll*, parará todo tipo de sonidos (se puede añadir un sonido de muerte) y al cabo del tiempo que se desee, desaparece.

Código destinado a iniciar el ragdoll tras la muerte.



OPTIMIZACIÓN

La optimización para Vindicator ha sido un punto importante y donde se ha requerido de una búsqueda extensa de técnicas y conocimiento específico.

Es muy importante cuando se quiere realizar un proyecto de estas características, empezar a abarcar la optimización desde el comienzo.

Lo primero de todo, es definir cuál va ser el *target* que apunte el proyecto. En este caso, el proyecto apunta a *next gen* y *high end*. El primer problema fue precisamente, realizar un producto pensado para nuevas generaciones, pero sin tener un equipo de nueva generación. Esto, a pesar de retrasar el desarrollo en ciertas partes, ha ayudado a expandir el proceso de optimización, llegando a ser jugable con un equipo más modesto, siendo el mínimo recomendable (el equipo usado para producción) sería una gráfica GTX 970 o similar.

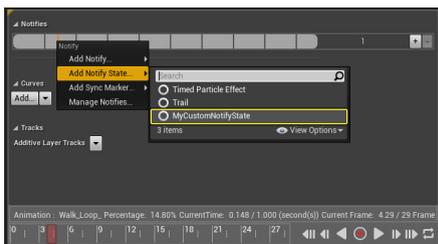
Para consultar el listado de técnicas y pasos en cuanto a la optimización, consultar el anexo **Memoria ampliada**.

SONIDOS

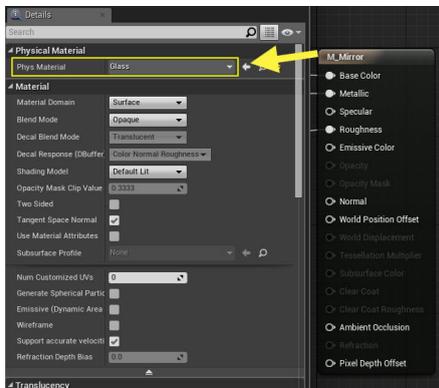
Uno de los elementos más importantes a la hora de la inmersión es el sonido, en este caso en particular, el que emite el personaje desplazado por el entorno. Esto genera mayor dinamismo e interactividad del personaje con los escenarios, situándose en un mundo más realista y creíble para el jugador.

En este caso, la forma que se ha usado para detectar cuándo emitir sonidos como los pasos ha sido usando las propias animaciones.

Hay otras maneras, por ejemplo, situar un *socket* que en cuanto detecte colisión emita sonido; sin embargo, de esta manera, se pueden desarrollar problemas derivados de las mismas colisiones, a parte, de que no se podría variar dependiendo de la acción (sería el mismo sonido de los pasos que si cae desde una altura). Por estos motivos, la manera de proceder fue mediante *animation notify* (Animation Notices, s.f.), que se basa en colocar en el frame concreto que precise la aparición del sonido, situar una notificación de sonido a la que posteriormente anclar una pista de audio. De esta manera, en cada animación se pueden tener distintas notificaciones, a las cuales adjuntar sonidos distintos.



Proceso de crear un *animation notify* en un punto específico de la animación.



Captura sobre la selección de physical materials y aplicación dentro de un material.

Una vez determinado cuando van a aparecer los sonidos, se configuran los distintos *physical materials* (Physical Materials User Guide, s.f.). Esta es una opción que tiene Unreal Engine para, de forma rápida y sencilla, poder adjudicar a los objetos o materiales que se hayan creado previamente, un *tag* concreto.

Esto permite que, a la hora de la aplicación de efectos, sonidos y *decals* procedentes de disparos (donde el sonido y *sprites* sean distintos si disparas un material de madera o metal) dependiendo de la superficie.

En este caso, los *physical materials* han sido utilizados para reproducir distintos sonidos al caminar jugador dependiendo de la superficie por donde camina así como distintos *decals* (si es tierra, aparece un pequeño sistema de partículas simulando que levanta polvo; también deja una huella que desaparece con el tiempo), distinto sonido de impacto al disparar al igual que distintos *decals*.

Sonidos como el de los zombies, varían según su estado y los clips de audio han sido sustraídos a partir del trabajo de Kevin Sherwood en *Call of Duty* (Kevin Sherwood, s.f.).

3.2 PRODUCCIÓN: INTERFAZ USUARIO



Ejemplo de menú principal en los videojuegos Red dead redemption 2 y Resident evil village.



Captura del menú creado para Vindicator.

Debido a la necesidad de integrar ciertas características como un control de los ajustes en la calidad visual para permitir mejores tests, se contempló la opción de realizar un menú principal para Vindicator.

En cuanto a los referentes, ha sido inspirado por títulos ya *mencionados anteriormente como el Red dead redemption, otros como el Resident evil village, Assassins creed Valhalla* entre otros.

El menú principal consta de un *gif* sacado a partir de un render real del mapa, el sonido capturado por el render y varios botones con sonidos. La razón de separar la imagen del sonido en vez de usar directamente un video es por razones de flexibilidad poder mutear, subir audio por separado.

Se pretende crear un menú simple, evitando la mayor cantidad de botones en pantalla y que mantuviera la estética. Cabe mencionar que los sonidos utilizados para la interacción de los botones, tanto el *Hover* como el *clic*, fueron realizados por *foley*.

En cuanto al menú de opciones, se han configurado distintas opciones gráficas, todo ello destinado al primer lanzamiento de prueba para poder tener un mayor control.

Se ha configurado en un solo botón todas las funciones destinadas a mayor calidad, a parte de otros efectos y el DLSS. Todo esto para ahorrar tiempo.



Captura del menú de cambios gráficos para vindicator.

En cuanto al sonido referente del menú, a parte de lo mencionado con el sonido del render y los efectos de sonido de los botones, se le ha añadido la banda sonora. Al darle a play, reproduce una pista de sonido mientras transiciona hacia la pantalla de carga.

La pantalla de carga se basa en un nivel por separado que en cuanto entras, reproduce un video y sonido mientras carga de forma asíncrona el nivel principal. Cuando ha terminado de cargar, realiza un fundido a negro y entra en el nivel. Para la demostración, directamente se ha dejado el video completo.



Captura de la pantalla de carga creada para Vindicator.

El video consta de una serie de renders de todo el mapa construido.

Debido a no poder realizar la cinemática inicial, igualmente, se decidió implementar el discurso que daba el protagonista. Este discurso comienza al final de la pantalla de carga y termina al entrar al juego. Está recogido de la famosa película Pulp Fiction, de la escena final donde el actor Samuel L. Jackson reproduce la cita célebre.

Se cogió el audio, se limpió y editó para simular el espacio donde se recita.

3.3 PRODUCCIÓN: GUIÓN

Para la historia principal, sin entrar en detalles narrativos (más bien se pretende abordar desde un punto de vista de estructura literaria), se piensa desarrollar la posible historia según la estructura aristotélica (Planteamiento, nudo y desenlace). Esta ha sido la forma más antigua y recurrente en la actualidad a la hora de organizar las historias (pues era ya estudiada por los griegos y recogida en mucho cine del Hollywood actual). De esta forma, se busca un formato sólido de principio a fin.

También se ha considerado importante fusionarlo con la estructura conocida como el camino o el viaje del héroe; esta organización fue explicada por primera vez por el mitógrafo Joseph Campbell en 1949 en su libro *El héroe de las mil caras*.

Para más información sobre el desarrollo del guión completo, consultar el anexo ***Memoria ampliada***.

3.3 PRODUCCIÓN: RENDERS

A continuación, los enlaces y una muestra de capturas de pantalla sacadas *in game* sin edición posterior :

[▶ Vindicator | Environment design | In game rendering](#)

[▶ Vindicator | Environment design | In game rendering](#)

[HTTPS://YOUTU.BE/LU9KBRDY74U](https://youtu.be/LU9KBRDY74U)

[HTTPS://YOUTU.BE/JPJLNCG4FOC](https://youtu.be/JPJLNCG4FOC)

















Para más imágenes, consultar *Libro de arte* en el *Anexo II*.

4. CONCLUSIONES

Tras finalizar este trabajo, atendiendo a los objetivos planteados al comienzo, se puede concluir que todos ellos han sido cumplidos, incluso pudiendo abordar muchos más aspectos de los propuestos.

Se ha conseguido abordar el escenario de una forma explorable mediante un personaje y su sistema de locomoción creado, resultando a su vez, a nivel técnico y visual, próximo a un juego AAA. En cuanto a la estética, se ha conseguido el objetivo de aproximarse a la época siguiendo los referentes pictóricos y la creación de contenido siguiendo fielmente reglas reales y de fabricación de la época a representar, así como una atmósfera, iluminación y sonido en consonancia.

En cuanto al segundo objetivo, se ha conseguido optimizar el rendimiento mucho más de lo esperado en los parámetros concretados anteriormente.

Finalmente destacar que se ha integrado y desarrollado los conocimientos adquiridos durante el grado y expandiendo a aspectos más técnicos y profundos en el campo profesional de environment artist así como muchos otros entre los que se encontraría programación, sonido, diseño de interfaz de usuario etc. El resultado final ha sido muy por encima de las expectativas personales, tanto por resultado como por la cantidad de conocimiento adquirido, técnicas y *workflow*.

5. BIBLIOGRAFÍA

About BoxCutter - boxDocs. (n.d.). Boxcutter-Manual.readthedocs.io.
<https://boxcutter-manual.readthedocs.io/en/latest/>

Advanced Locomotion System V4 in Blueprints - UE Marketplace. (s.f.). Unreal Engine. Retrieved June 22, 2022, from
<https://www.unrealengine.com/marketplace/en-US/product/advanced-locomotion-system-v1?sessionInvalidated=true>

Andy Chalk published. (2022, April 5). CD Projekt explains why The Witcher 4 is using Unreal Engine 5. *PC Gamer*.
<https://www.pcgamer.com/cd-projekt-explains-why-the-witcher-4-is-using-unreal-engine-5/>

Animation Notifies. (n.d.). Docs.unrealengine.com. Retrieved June 22, 2022, from
<https://docs.unrealengine.com/5.0/en-US/animation-notifies-in-unreal-engine/>

Curvature - Substance 3D Painter. (n.d.). Substance3d.adobe.com.
<https://substance3d.adobe.com/documentation/spdoc/curvature-142213459.html>

Documentation – Arm Developer. (s.f.). Developer.arm.com.
<https://developer.arm.com/documentation/102696/0100/Texture-channel-packing>

Descubre la importancia de Nanite. (2021, July 2). UT-HUB.
<https://www.ut-hub.com/nanite-unreal-engine-5/>

Early 20th Century Fabric Glossary. (2020, December 17).
<https://www.sewdecadesago.com/?p=1318>

Encyclopaedia Britannica. (2021). Industrial Revolution. In *Encyclopedia Britannica*.
<https://www.britannica.com/event/Industrial-Revolution>

Grimshaw, Atkinson. (n.d.). Museo Nacional Thyssen-Bornemisza. Retrieved June 22, 2022, from <https://www.museothyssen.org/coleccion/artistas/grimshaw-atkinson>

Kevin Sherwood. (n.d.). Call of Duty Wiki. https://callofduty.fandom.com/wiki/Kevin_Sherwood

Leonardo Iezzi - Lead Level Artist. (s.f.). Leonano.com. Retrieved June 22, 2022, from <https://leonano.com>

Lumen Global Illumination and Reflections. (s.f.). Docs.unrealengine.com. <https://docs.unrealengine.com/5.0/en-US/lumen-global-illumination-and-reflections-in-unreal-engine/>

MODULAR BUILDINGS: TIPS AND TRICKS. (2019, OCTOBER 21). 80.LV. <https://80.lv/articles/modular-buildings-tips-and-tricks/>

Physical Materials User Guide. (n.d.). Docs.unrealengine.com. Retrieved June 22, 2022, from <https://docs.unrealengine.com/5.0/en-US/physical-materials-user-guide-for-unreal-engine/>

Stone, C. J. [from old catalog, & The Library of Congress. (1887). Stone's paramount-cutter; a system for cutting garments, based upon scientific principles, including the self-varying shoulder, in connection with the division of the breast-measure. In *Internet Archive*. Chicago, Meyer & brother. <https://archive.org/details/stonesparamountc01ston/page/24/mode/2up>

Wikipedia, la enciclopedia libre. (2012). Wikipedia.org. <https://es.wikipedia.org/wiki/>

Un nuevo Tomb Raider está en desarrollo con Unreal Engine 5 por Crystal Dynamics. (s.f.). Vandal. Retrieved June 22, 2022, from <https://vandal.elespanol.com/noticia/1350752747/un-nuevo-tomb-raider-esta-en-desarrollo-con-unreal-engine-5-por-crystal-dynamics/>

6. ANEXOS

Como apoyo a este documento, han sido creados dos anexos complementarios. El primero, Memoria ampliada, donde se abordan temas más específicos o que por su longitud y necesidad de apoyo visual, no podía incrustarse en esta memoria. Destacar que cuenta con una serie de consideraciones personales en cuanto al uso de técnicas descritas, tanto consejos para aplicaciones prácticas, como opiniones personales sobre las mismas.

El segundo anexo, Libro de arte, es un catálogo de imágenes que recopila las distintas partes del escenario con iluminaciones distintas, renders del personaje y armas.

Por último, incluir nuevamente videos del proyecto:

<https://youtu.be/eJHPz7tKwAM>

<https://youtu.be/IIx4zaEiu8A>

<https://youtu.be/XR7eMQ6z93o>

<https://youtu.be/Lu9KBRDY74U>

<https://youtu.be/jpJIncG4fqc>