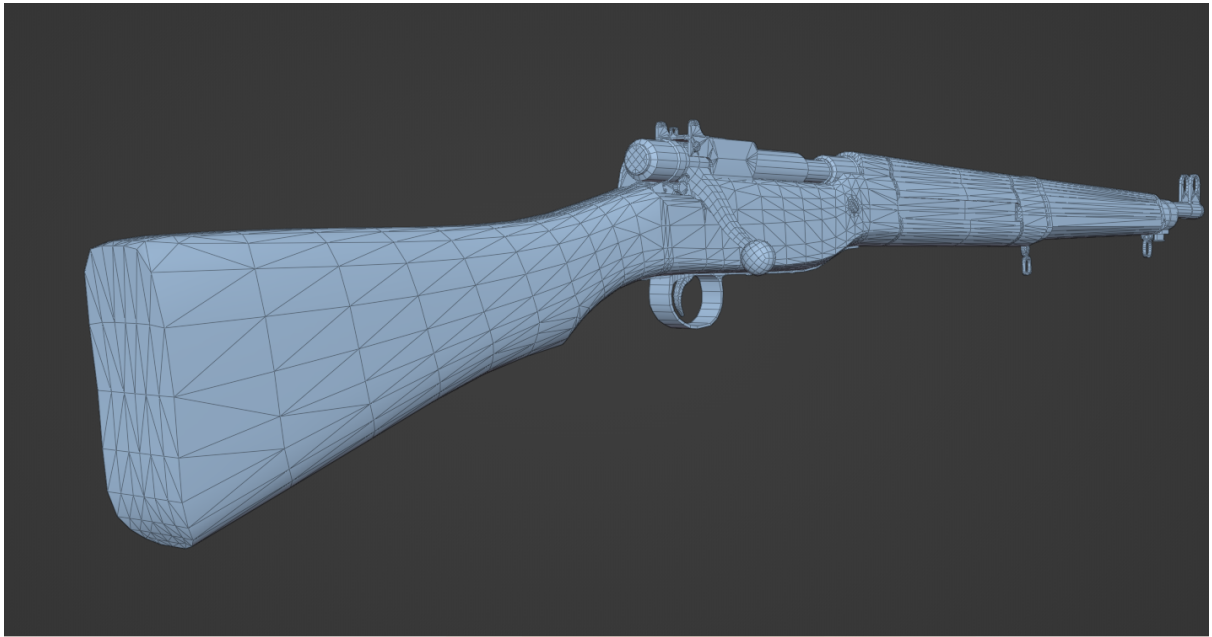


Índice

- Modelado
 - **Props**
 - **Armas**
- Animaciones
 - **Armas y accesorios**
 - **Ajustes**
- Motor videojuego
 - **Complex materials, POM**
 - **Partículas**
 - **Escenarios desechados**
 - **Sonidos**
- Iluminación
 - **Iluminación en tiempo real**
 - **Post Procesado**
- Optimización
- Guión

Props : Armas

A pesar de ser un elemento que no aparece en todo momento en pantalla y que tampoco se visualiza en primer plano, debido a la posibilidad de estar en cinemáticas, han sido creadas a mayor resolución que el resto; esto también permite plantear el juego en primera persona.





Debido a la necesidad de mayor resolución, las armas fueron separadas en 4 materiales distintos, permitiendo mayor superficie en las UV, de lo que más adelante se abordará en detalle.

Consideraciones: Más imágenes disponibles en el apartado **Libro de Arte**.

Animaciones: Armas y accesorios

En el caso de las armas, el proceso fue más sencillo. Simplemente se importó el arma con los huesos que viene como ejemplo del ALS en Blender, se eliminó la malla, se posicionan los huesos en el lugar correspondiente en el nuevo arma y se aplica el esqueleto a la malla.

Más tarde en Unreal, se añaden distintos sockets, de donde spawnear los proyectiles, sonidos etc.



Animaciones: Ajustes

Debido a usar un sistema con animaciones precocinadas, es necesario adaptar las animaciones al contexto del videojuego; En este caso, se optó por modificar animaciones como la de apuntado a una mano con revolver, donde se han seguido referencias de juegos actuales como Red Dead Redemption 2, juego ambientado en la América de 1896, siguiendo muchos de los referentes en cuanto al cine Western.



Se ha modificado también la velocidad a la que se desplaza el personaje, se han limitado algunas acciones (con el objetivo no de tener un gameplay dinámico y rápido, sino con objetivo de servir al objetivo de poder mostrar el entorno).

También se ha modificado ligeramente la forma de andar, siguiendo referencias como la serie de Netflix *Peaky Blinders*, donde los protagonistas tienen una forma característica de caminar, encorvando la espalda hacia delante y con los brazos separados...



Motor de videojuego: Complex materials

Dentro de las posibilidades que ofrece Unreal Engine, está la creación de materiales complejos que ayuden al desarrollo artístico del proyecto. Algunos como el vertex paint, que permite tener varios materiales aplicados directamente en el motor sin necesidad de pasar por softwares de texturizado, pudiendo cambiarlo al gusto.

Como añadido, debido a que no se puede aprovechar Nanite ni darle una geometría concreta a la malla, para poder simular mayor tridimensionalidad y profundidad de los materiales, se usan técnicas como el Bump Offset.

A diferencia de la técnica de modelado de desplazamiento, que mueve geometría existente, esta forma permite mediante un mapa en escala de grises, usualmente el Heightmap, simular una geometría desplazada mediante las UV sin necesidad de geometría extra, siendo directamente renderizado en GPU. Esta forma artificial de generar geometría, tiene sin embargo, la limitación dada por su naturaleza: En caso de que el jugador mire desde los laterales de la malla, queda en evidencia que no hay geometría y se ve plano.

También cabe destacar otra limitación, y es que al no ser geometría real, no va a tener colisión, con lo que el jugador traspasará las partes levantadas o flotará.

Sin embargo, la posibilidad de combinar esto con el Vertex Paint, permite añadir mucho más detalle y poder dividir los materiales entre capas (donde dependiendo del material, sobresaldrá más o menos).

Para el proyecto, también se ha utilizado otra forma de Bump, llamada Parallax Occlusion Mapping. Esta forma es similar al Bump Offset, con la diferencia en el tratamiento de la luz. Con el Bump Offset, no hay variación dependiendo de dónde venga la fuente de luz. Esto es corregido con el POM, dando mayor realismo pues interactúa en mayor medida con la iluminación.



Destacar que el uso de esto, conlleva mayor carga de la gpu, con lo que es recomendable limitarlo a ciertas partes y darle una distancia a partir de la cual, deje de actuar para aliviar la carga de renderizado. También añade Render passes , con lo que supone una mayor cantidad de Draw calls.

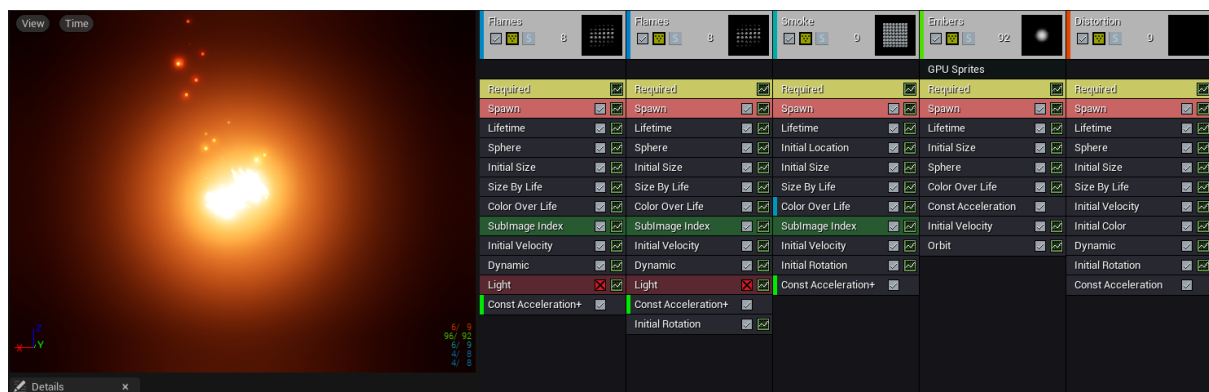
Motor de videojuego: Partículas

Otro de los aspectos fundamentales en la creación de videojuegos, es el apartado de VFX o efectos visuales.

Generalmente, esto se consigue mediante partículas o sprites. Estos son usado para elementos como fuego, humo, chispas, disparos...

Unreal Engine 5 cuenta con Niagara, que es un sistema VFX para next-gen, que permite gran funcionalidad y flexibilidad sin necesidad de programación.

La forma creación de estas se basan en spawnear una textura, dándole un radio, un número inicial, una velocidad etc. y, a partir de ahí, darle el movimiento que va a seguir esa textura (u objeto).



Debido al coste de renderizado que pueda suponer, es aconsejable plantear si va a ser mediante GPU o CPU. La diferencia principal, a grandes rasgos, es que si se renderiza mediante GPU, es que cuando la partícula esté en pantalla, es cuando va a renderizarse. Por el contrario, en la CPU, va a estar todo el rato activa ocupando un proceso, aparte de consumir GPU cuando vaya a mostrarse en pantalla.

Normalmente, partículas como la lluvia, que suelen spawnear en un radio concreto alrededor del jugador y a una altura concreta, es necesario que se calcule por la CPU, pues de no hacerlo, únicamente se verían de mirar al cielo; es decir, necesita estar ejecutándose incluso si no aparece en pantalla.

Motor de videojuego: Escenarios desechados

La idea principal del escenario, debido a la complejidad de crear un mundo abierto que no esté vacío y por las limitaciones de tiempo, se optó por acotar a una zona de lo que sería el mapa jugable; esta consta de 3 partes principales: La iglesia, el pueblo y la fábrica, todo ello sin contar las partes que conectan cada zona, pues igualmente han de ser importantes.

Las primeras creaciones y block out que se realizaron, tras una revisión del mapa y estando relativamente avanzado, carecían de verticalidad, y daban un aspecto a una zona demasiado plana. Se reconstruyó, situando la iglesia en una zona elevada, otorgando más protagonismo y presencia. Se incrementó la distancia con el pueblo y a este último, se le aumentó el tamaño.



Como se puede apreciar, el camino que conectaba el pueblo y la iglesia era mucho más corto, pudiendo visualizarse esta última en el horizonte.



Motor de videojuego: Sonidos

Uno de los aspectos principales en los videojuegos es el sonido, ya sea banda sonora, música ambiental, efectos de sonidos, locuciones de personajes...

De esta manera, el diseño de sonido en Vindicator se divide en varios apartados:

Leit motiv del juego, que ha sido compuesto para este proyecto;

Sonido ambiente, que en este caso, sigue a la meteorología representada, la cual es una tormenta incluyendo relámpagos. Cabe añadir otros sonidos como el movimiento de la vegetación, el impacto de las gotas de agua...

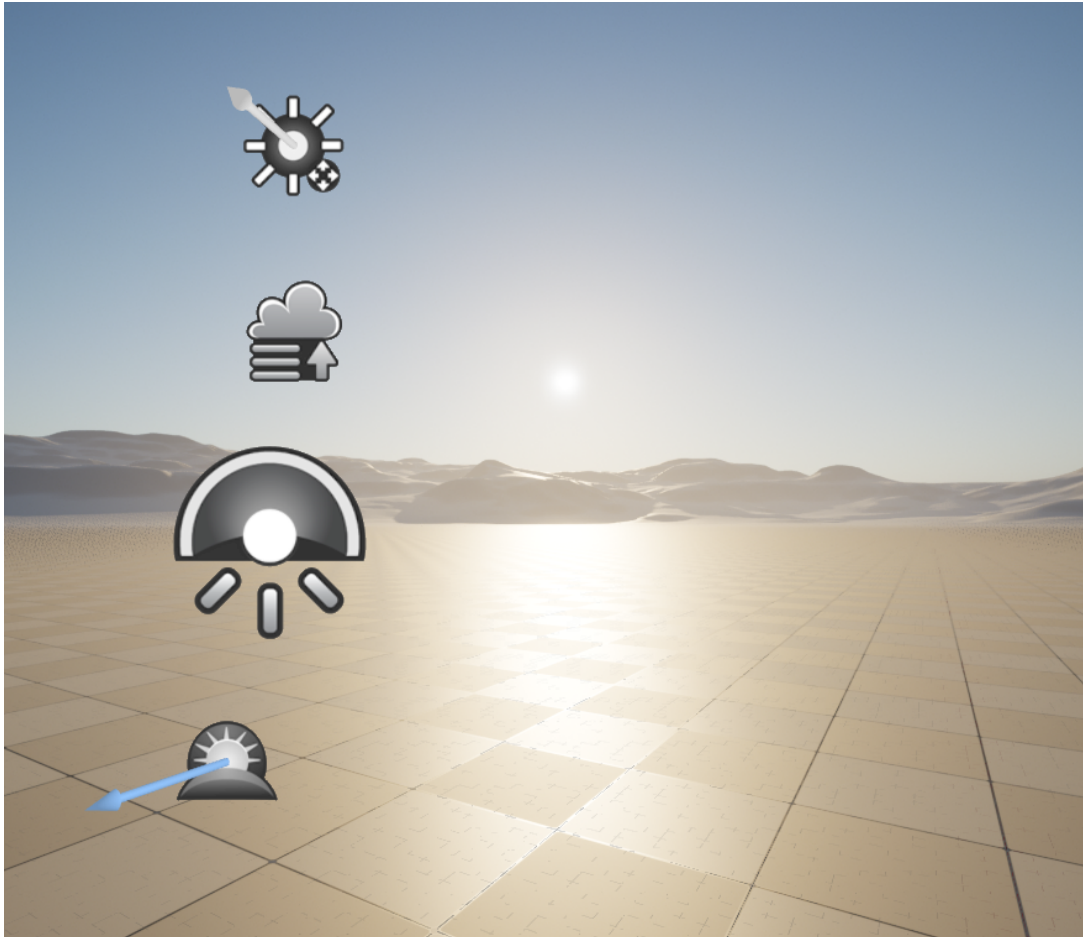
Efectos de sonidos, los cuales el jugador puede percibir durante el gameplay. Aquí se hace la distinción entre, los producidos por el personaje, los zombies y los del entorno.

Los del personaje son todos aquellos propios de la interacción con el entorno, siendo el sonido de las pisadas de distintas superficies; estos sonidos han sido capturados de otros juegos, como *Assassins Creed Valhalla*, *Red dead redemption 2*. Posteriormente, se han importado a Adobe Audition, donde han sido limpiados y posteriormente, integrados. Como ya se ha comentado, dependiendo de la superficie pisada, se emitirá un sonido u otro, y cada tipo de superficie pisada, cuenta con más de 4 pistas de audio que se reproducirá de forma aleatoria para mayor variedad. La razón por la cual coger el sonido de juegos existentes, es debido a las diferencias entre el sonido en videojuegos y otros formatos como animación o tv. En videojuegos, la espacialización y atenuación te lo da la distancia de la cámara conforme a la fuente del sonido y su alrededor. De esta manera, los sonidos que se puedan extraer de los videojuegos suelen ser más puros, al no estar directamente editados como en otros formatos.

En cuanto a los zombies, el sonido al caminar por las distintas superficies es aprovechado de los del personaje; En cuanto a los vocales de estos, gritos etc, han sido nuevamente extraídos de un videojuego, concretamente de la saga *Call of Duty*. La razón radica en ser sonidos muy característicos e icónicos en el mundo de los videojuegos. El sonido fue producido por Kevin Sherwood, diseñador de sonido para Treyarch (https://callofduty.fandom.com/wiki/Kevin_Sherwood).

Por último, los diferentes sonidos en el entorno que pueden ser escuchados por el jugador, por lo general, se usan de complemento a un sistema de partículas, como por ejemplo fuego, donde se le añade sonido de llamas y, dependiendo de qué esté ardiendo, cambia el sonido, por ejemplo, en una parte del mapa hay una carreta de madera ardiendo, a esta se le añade un fuego más típico de hoguera, con crujido de madera etc. Otros sonidos como el de las telas en movimiento, sonido de moscas...

Iluminación: Iluminación en tiempo real



La forma que se ha construido la iluminación de Vindicator ha sido mediante los siguientes elementos:

Luz direccional, fuente de luz principal que actúa a modo de sol o luna.

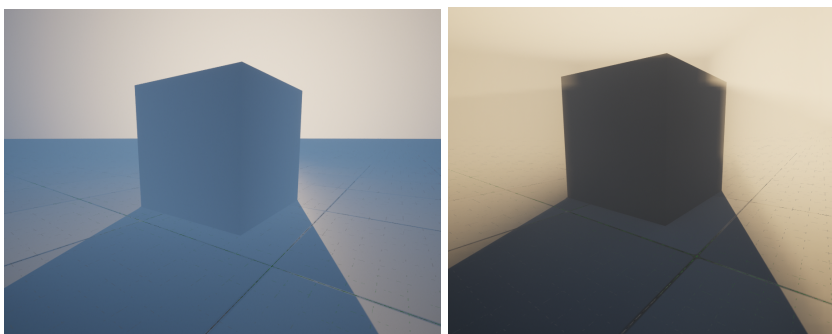
Sky atmosphere, es una técnica de representación del cielo y la atmósfera físicamente basado en el cielo real. Este componente actúa junto con la luz direccional, simulando los cambios de luz y color dependiendo de la posición del sol; a su vez, actúa con la niebla de altura exponencial, y permite cierta flexibilidad como nivel de absorción de luz, tamaño de la atmósfera etc...



<https://docs.unrealengine.com/4.27/en-US/BuildingWorlds/FogEffects/SkyAtmosphere/>

Exponential height fog, que a diferencia de la niebla tradicional, esta permite transicionar hacia qué altura tener mayor densidad. Por defecto, te da la posibilidad de teñir el fog con los colores que se desee, uno para el hemisferio que se sitúa mirando a la fuente de luz, y otro para el opuesto. Como se ha comentado, puede actuar en consonancia con el Sky Atmosphere, de manera que si se selecciona en ambas opciones el color negro (es decir, el valor 0), el fog adaptará el color que adquiera el cielo dependiendo de su posición.

Otra característica importante es la capacidad de activar la parte volumétrica. La forma de entender el comportamiento de esta característica, es analizando cómo funciona el fog; este es un proceso aditivo el cual es añadido y mezclado a la imagen dependiendo la distancia y densidad, lo que significa, que partes del mapa donde no haya luz, estarán parcialmente iluminadas por la niebla. Cuando se activa la opción de niebla volumétrica, en vez de simplemente añadir el color por toda la pantalla, pasa a hacerlo en las partes de luz, creando los famosos God Rays.



Se puede controlar la absorción de luz y demás parámetros.

La opción de controlar la densidad también puede ser ajustada directamente en las fuentes de luz, para mayor flexibilidad, permitiendo diferentes zonas con distinta densidad.

<https://docs.unrealengine.com/4.27/en-US/BuildingWorlds/FogEffects/VolumetricFog/>

Nubes volumétricas, que al igual que el fog, actúa de la misma forma, pudiendo poner a lo largo del cielo masa de nubes que dispersan la luz, provocando cambios naturales de iluminación similares a la vida real. A su vez, pueden ser animadas.



Un juego ya mencionado anteriormente que hace un buen uso de nubes volumétricas, es el *Red dead redemption 2*.



<https://docs.unrealengine.com/4.27/en-US/BuildingWorlds/LightingAndShadows/VolumetricClouds/#:~:text=The%20Volumetric%20Cloud%20component%20is,they%20need%20for%20their%20projects>

Sky light, el encargado de capturar las distintas partes del nivel y aplicarlas en forma de iluminación.

<https://docs.unrealengine.com/4.27/en-US/BuildingWorlds/LightingAndShadows/LightTypes/SkyLight/>

Iluminación: Post procesado

El post-procesado en los videojuegos es el proceso mediante el cual se aplican filtros y efectos a la imagen en un videojuego. Es muy importante, pues tienes la posibilidad de dotar de personalidad y singularidad al juego.

En Unreal Engine, la manera de aplicar efectos de postprocesado, es mediante el actor que puede ser añadido al nivel. Este puede escalarse para ajustarse a una zona concreta o extenderlo infinito, lo que significa, que todos los efectos se aplicarán en todo el nivel.

A continuación, el desglose de los distintos apartados que han sido modificados:

Bloom, que es un fenómeno que ocurre en el mundo real y que dota de mayor realismo al juego con un coste de renderizado bajo. La luz cuando se encuentra sobre un fondo oscuro, se extiende sobre él.

<https://docs.unrealengine.com/4.27/en-US/RenderingAndGraphics/PostProcessEffects/Bloom/>

Exposición, pues al tratarse de una escena con bastante contraste y por la noche, es necesario que haya cierto nivel de autoexposición, para que se ajuste dependiendo de la zona.

Aberración cromática, que sucede en los contornos de los objetos situados sobre un fondo brillante y que separa los colores RGB. Se usa principalmente para desenfocar los laterales de la pantalla y dejar en el centro una imagen más nítida.



<https://docs.unrealengine.com/4.26/en-US/RenderingAndGraphics/PostProcessEffects/SceneFringe/#:~:text=Scene%20Fringe%20is%20a%20chromatic,the%20edges%20of%20the%20image.&text=Sce ne%20Fringe%20settings%20can%20be,in%20the%20Post%20Process%20Settings.&text=A%20scaling%20factor%20that%20controls%20how%20much%20shifting%20occurs.>

Máscara de suciedad, que es notoria cuando el jugador mira directamente a una luz, apareciendo en pantalla manchas que podría tener una lente de cámara real.

Flash de lente, que ocurre nuevamente cuando el jugador mira directamente a un punto de luz u objeto emisor. Estos efectos están enfocados a generar mayor inmersión en el juego.

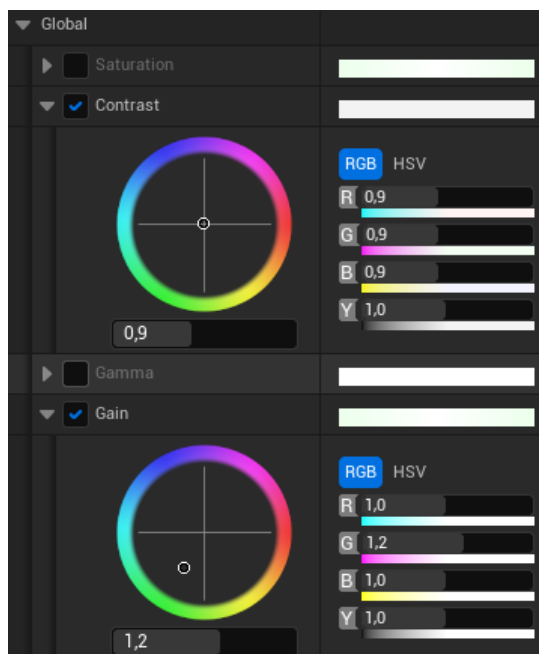
Viñeteado, que consiste en oscurecer los bordes. Junto con la aberración cromática, sirve para enfocar al jugador al centro de la pantalla.

Profundidad de campo, usado para desenfocar progresivamente hacia el horizonte.

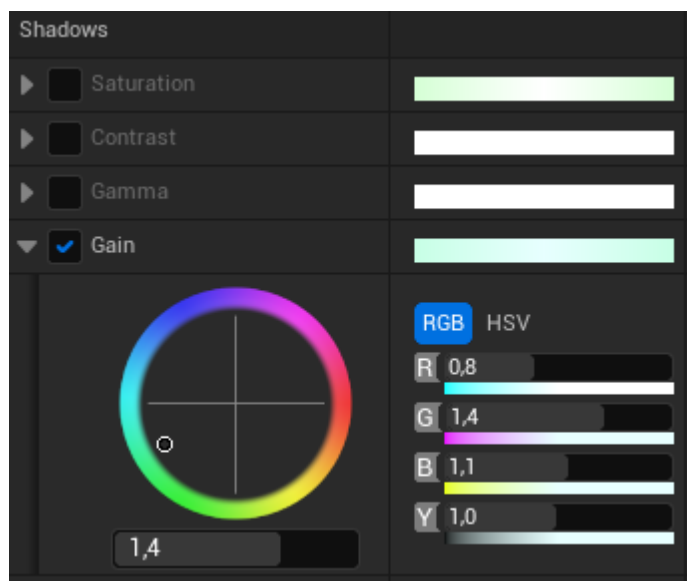
En cuanto a cuestiones de color, que es lo más característico del título Vindicator y donde más énfasis se le ha puesto, la idea era reconstruir el ambiente en las obras de John Atkinson, pintor inglés del siglo XIX.



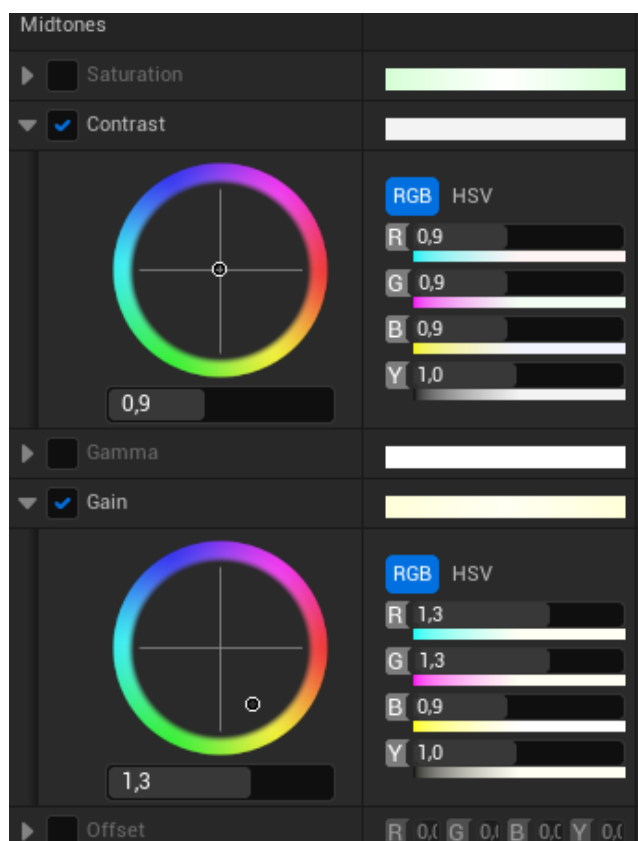
Los parámetros y sus medidas son los siguientes:



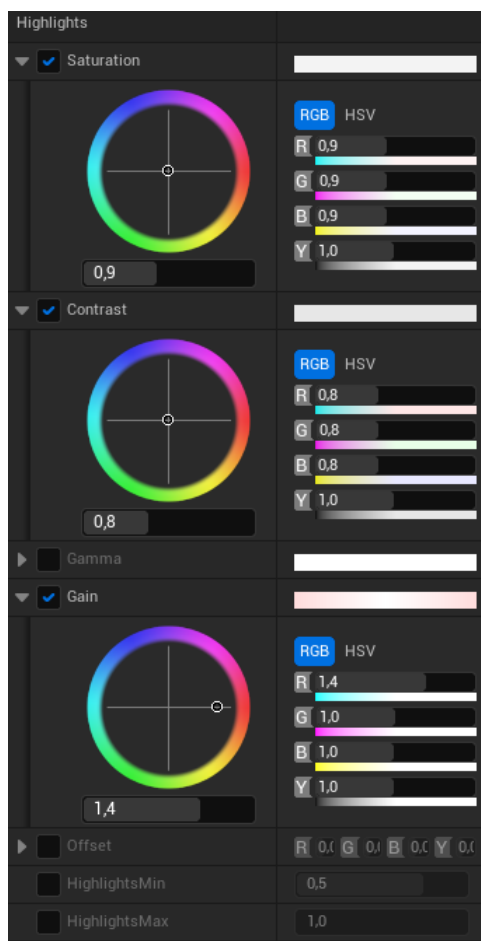
Para el color global, se ha reducido el contraste 0.1 y se ha subido la ganancia de verde 0.2. La reducción ligera de contraste ayuda a unificar la escena y la ganancia de verde para dotar del tono verdoso característico.



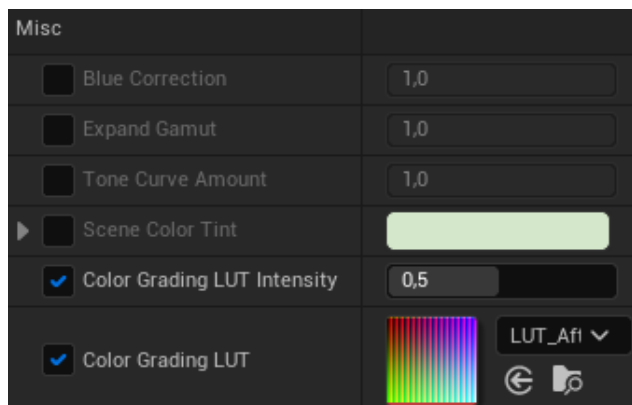
Para las sombras, se ha reducido la ganancia del rojo en 0.4, se ha subido la del verde y azul 0.4 y 0.1 respectivamente. Destacar que se hicieron diversas pruebas, como dejar las sombras con un tono más azulado, pero finalmente tras diversas encuestas a compañeros de clase, se decidió mejor por el verde.



Para los tonos medios se ha reducido el contraste 0.1 para mayor claridad y se ha aumentado la ganancia de rojo y verde en 0.3 mientras que el azul se ha disminuido en 0.1. Esto actúa como transición entre las sombras y los claros, como se verá más adelante.



En los claros se ha reducido la saturación en 0.1, reducido el contraste en 0.2 puesto que el bloom ya se encargaba de potenciar los claros; se ha subido la ganancia del rojo en 0.4, haciendo los claros más cálidos.



Por último, se hizo un render de la imagen y en Adobe Photoshop se terminó de retocar. Al término, se procesó la imagen para elaborar un LUT y se importó a Unreal Engine. Se le redujo la intensidad, puesto que el perfil de color era distinto.

Para finalizar, se le añadió el efecto de grano de película, para homogeneizar la imagen y dar un aspecto más vintage.

Optimización

Las diferentes fases o técnicas de optimización son las siguientes:

Para los modelos en 3D, se ha usado la tecnología de Nanite, que es actualmente, lo mejor para optimizar a nivel poligonal (Si está permitido). A su vez, se ha creado un volumen de *Distance Culling*, que se basa en hacer desaparecer los objetos a partir de cierto tamaño a cierta distancia. Esto es útil para que la gráfica no intente renderizar objetos muy pequeños a partir de cierta distancia.

Para la parte de la vegetación, que es siempre lo más costoso debido al *Overdraw* que sucede por las texturas enmascaradas. Por lo general, la vegetación son modelos simples que usan una textura con transparencia; el problema que surge, es que la superposición de alfas es calculada por cada frame, siendo un proceso muy costoso. Para ello, se ha decidido usar el early Z pass, que esto funciona al guardar la distancia de la escena para cada píxel en un pre-pass, y luego, cuando ocurre la representación real, se pueden omitir las primitivas ocultas. Esto es usado en todos los motores de videojuegos para ganar mucho rendimiento, sobre todo en títulos que usan mucha vegetación.

Como la vegetación no está actualmente soportada por Nanite, la forma clásica de ahorrar rendimiento en los modelos, es mediante los LOD (Level of Detail). Esto se basa en, dependiendo de la distancia a la que se encuentre un modelo u objeto, utilice una malla decimada del mismo. Unreal Engine permite dentro del motor controlar la cantidad de decimación y la distancia que cambia. A su vez, te permite tener distintos controles, como que a partir de cierto LOD, deje de emitir sombras, utilice una textura sin alfas...

Otro de los apartados que más consume, es la iluminación. El cálculo en tiempo real de las luces y sombras es algo que hay que tener cuidado a la hora de crear escenas, pues una mala praxis supone hundir el rendimiento.

Las diferentes optimizaciones han sido: Reducir el número de Shadow Cascades, que por defecto, vienen 4. Esto supone que calcule las sombras 4 veces para mayor resolución, y se ha reducido a 2. Posteriormente, se ha configurado para crear algo similar al LOD de los modelos, pero para las sombras, teniendo la mayor resolución en las sombras cercanas y de baja resolución en las lejanas; incluso a partir de cierta distancia, se ha configurado para que las sombras más lejanas dejen de calcularse y use *Distance Field Shadows*, un tipo de sombra más barata basado en la simplificación de los modelos a partir de *distance field meshes*.

También, se ha configurado las *Contact Shadows* para simular sombras falsas por ejemplo, en vegetación donde por su nivel de LOD, dejan de emitir sombras dinámicas.

A nivel de uso de luces, se ha evitado el solapamiento de los puntos de luz, limitando la distancia de suavizado; se ha configurado un distance culling a cada luz por individual y con una distancia personalizada para que ilumine de forma progresiva y no aparezca de golpe; Se han utilizado luces donde no emiten sombra (en lugares específicos que se permite sin resultar muy obvio)

Cabe mencionar que Unreal Engine 5 trae un sistema de sombras, llamado Virtual Shadow maps, que es un sistema preparado para modelos de muy alto poligonaje y con alta resolución. Pero debido a las limitaciones del equipo y en pos de mejorar la optimización, se decidió no usarla.

A nivel de texturas, se han usado texturas con un máximo de tamaño en 4096x4096 (4K), aunque más tarde fuera a reducirse. Al igual que los modelos tienen el sistema LOD, las texturas usan los Mipmaps, donde reducen la resolución en base a la distancia o el tamaño en pantalla. Nuevamente, Unreal Engine tiene un sistema llamado *Texture Streaming*, que gestiona automáticamente este apartado, generando, para texturas 4K, 12 mipmaps que irán variando en cuanto al tamaño en pantalla. Para seguir afinando, en muchos modelos como la vegetación, se ha reducido todo lo posible el tamaño del mapa de opacidad (Por los problemas que presenta mencionados anteriormente) hasta 512px o, en algunos casos, 256px (siempre un múltiplo de 2 para que genere mipmaps). Luego, se ha intentado seguir la regla de que, la textura de Normal sea la mitad a la del Color, y el Roughness sea la mitad a la del Normal. Mencionar que esto depende también de la importancia del objeto, puesto que objetos que no van a estar cerca de la cámara en ningún momento, no necesita que se reproduzca a alta resolución.

Con los materiales, como ya se ha comentado anteriormente, es importante mantener reducido los draw calls. Para ello, todos los materiales parten de un Material padre, siendo todos instancias de este donde solo se necesita cambiar las texturas correspondientes. De esta manera comparten shader y hace mucho más liviano el trabajo.

Otro elemento muy costoso a nivel de renderización, es el fog local. Inicialmente, se había elaborado un sistema de partículas, pero más tarde, debido a problemas de rendimiento, se decidió crear un material con capacidad volumétrica que permite ser usado ampliamente sin apenas pérdidas de rendimiento. Esto fue debido al llamado *overdraw*, comentado anteriormente; cuando usas objetos o partículas como fog local, las capas forman parte de un proceso aditivo las cuales se van acumulando. Por eso es preciso añadir una distancia de renderizado.

Una de las herramientas utilizadas para apoyar el rendimiento, es el uso del DLSS de Nvidia. El DLSS es una técnica de supermuestreo surgida hace poco que hace uso de la potencia de las gráficas RTX en cuanto a inteligencia artificial; Lo que permite es, renderizando la imagen, por ejemplo, en 1920x1080, reescalar por inteligencia artificial y que reproduce en 4K, sin apenas perder calidad visual e incrementando el rendimiento drásticamente. Nuevamente, al no tener un equipo con una gráfica RTX, no se ha podido testear en profundidad, pero se ha dejado preparado y programado para su uso.

Una parte muy importante de la optimización mencionada numerosas veces con anterioridad, son los llamados draw calls. Los draw calls según la definición que da Unreal Engine, es un grupo de polígonos que comparten el mismo material. A la hora de optimización, es muy importante tenerlo en cuenta y mantener los mínimos draw calls posibles, puesto que pueden llegar a suponer mayor coste que el número de polígonos. Esto se basa en cómo el CPU lo maneja. Se transfiere una cantidad X de triángulos a la GPU por draw call y se aplicará el mismo estado de representación a estos triángulos. Entonces, por ejemplo, si hubiera la misma cantidad de draw calls, es más rápido transferir 1,000,000 de triángulos una vez que 100 triángulos 1000 veces. Porque la GPU tiene que detenerse y esperar a que la CPU transfiera el siguiente conjunto de información. Y esto aumenta de forma exponencial cuando cada objeto distinto tiene un material con un shader distinto.

Es por esto que se han utilizado Master materials o materiales padre de los cuales se han creado instancias; A pesar de cambiar los mapas de textura, el shader es el mismo, con lo que el cpu lee únicamente un material. Otra de las maneras de mantener los draw call bajos es mediante la agrupación de objetos, ya sea creando un actor con instancias o un Blueprint, que actúa de forma similar, pero permitiendo modificarlo.

<https://docs.unrealengine.com/4.27/en-US/BlueprintAPI/Components/InstancedStaticMesh/>

Por lo general, para videojuegos en consola y pc, se recomienda un número aproximado de entre 2000 a 3000 draw calls. En el caso de Vindicator, la mayoría de las veces se sitúa rozando los 2000 o menos.

Debido a problemas de potencia en el desarrollo, era necesario mediante el proceso de producción, poder trabajar cómodamente sin problemas de rendimiento para evitar cuelgues del equipo etc. Una de las formas que permite Unreal, es mediante la creación de *Sublevels* dentro del nivel principal. De esta forma se permite el trabajo en una zona concreta del mapa donde todos los objetos y assets se encuentran en ese sublevel en concreto. Esto actúa a modo de carpeta, pudiendo ocultarlo y permitiendo mayor comodidad en el montaje.

Un aspecto también fundamental a nivel de optimización, no en cuanto a la parte de rendimiento del proyecto, es en cuanto a optimización de workflow, metodología de trabajo, praxis... Debido a la magnitud inicial del proyecto, y al ser un trabajo individual, era necesario que los distintos apartados a abordar se realizaran de la manera más efectiva para poder cumplir los objetivos.

Es por esto que en cuanto al modelado de assets, se ha evitado en todo momento tener que realizar un remesh, sacando desde el comienzo una malla limpia sin problemas de sombreado y un flow correcto; o en el caso de necesitarlo, la utilización de plugins dentro de Blender para evitar exportarlo a otros softwares. En este sentido, se ha intentado reducir los objetos llevados a zbrush para realizar una proyección de detalles, llevándolos directamente a Substance Painter y realizándose allí los distintos detalles. Acompañado de distintos métodos comentados anteriormente como la creación de Smart materials que permiten reusarse de forma rápida...

Metodología

(transcripción de los diagramas de trabajo y cronología)

El trabajo dio comienzo con su fase de ideación; la idea del proyecto, motivada por títulos de videojuegos AAA tales como Days Gone, The Last of Us, Red Dead Redemption 2 entre otros, surgió como la posibilidad de realizar una maqueta de juego AAA de mundo abierto con mecánicas y dinámicas similares a la de los títulos mencionados. A su vez, serviría como proyecto profesional para portfolio.

Los primeros pasos del proyecto se dieron en la búsqueda de una época y estética que no haya sido muy explotada y con potencial para ofrecer un producto singular. Motivado por series actuales como Peaky Blinders, se decidió optar por ambientar el juego en la época de la revolución industrial de la Inglaterra del siglo XX. A su vez, se tomó la opción de elaborar una narrativa a partir del género zombies, puesto que las condiciones de vida de esa época junto con una expansión acelerada, daba paso a elaborar tramas complejas y con perspectivas en cuanto a recursos. Las primeras ideas viraban a la creación de una iglesia, de donde partía el personaje principal y su valuarte del mundo que le rodeaba; este saldría y se dirigiría a una ciudad donde la atravesaría hasta llegar a una fábrica, que sería la entrada y salida hacia los muelles de la ciudad. Tras brevemente definir cómo sería el mapa, se comenzó con la búsqueda de referentes en cuanto a entorno. Paralelamente se seguía trabajando en el desarrollo de la idea, definiendo al personaje, el lore en el mundo que se iba a construir. Se empezaron a buscar precedentes en el género de zombies para desarrollar cómo serían, su velocidad, si presentaban deformidades, búsqueda de información sobre enfermedades reales que pudieran respaldar un diagnóstico realista de un virus. Se desarrolló cuestiones relacionadas con el gameplay: sería un juego en tercera persona, que capacidades tendría el personaje, las armas que llevaría.

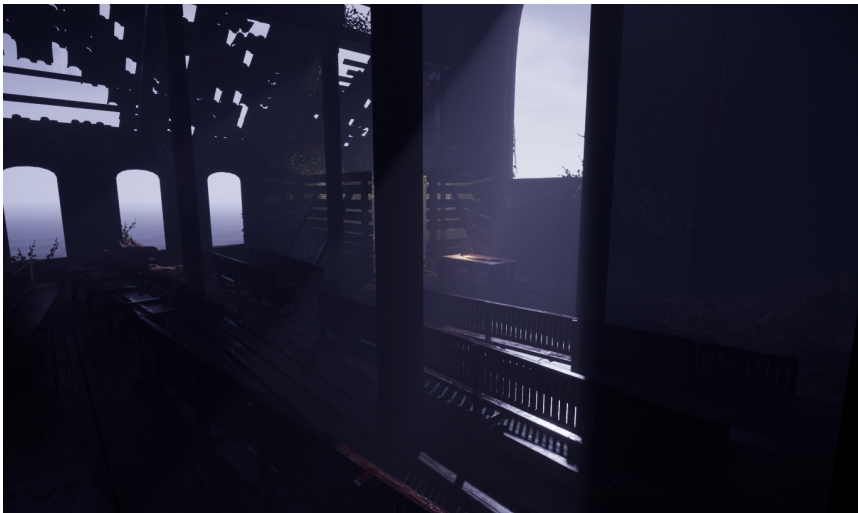
Destacar que en los tiempos donde se recorría esta serie de cuestiones, para contextualizar, estaba publicado el Early Acces de Unreal Engine 5, donde aparecían tecnologías revolucionarias y animaban a explotar al máximo el apartado artístico.

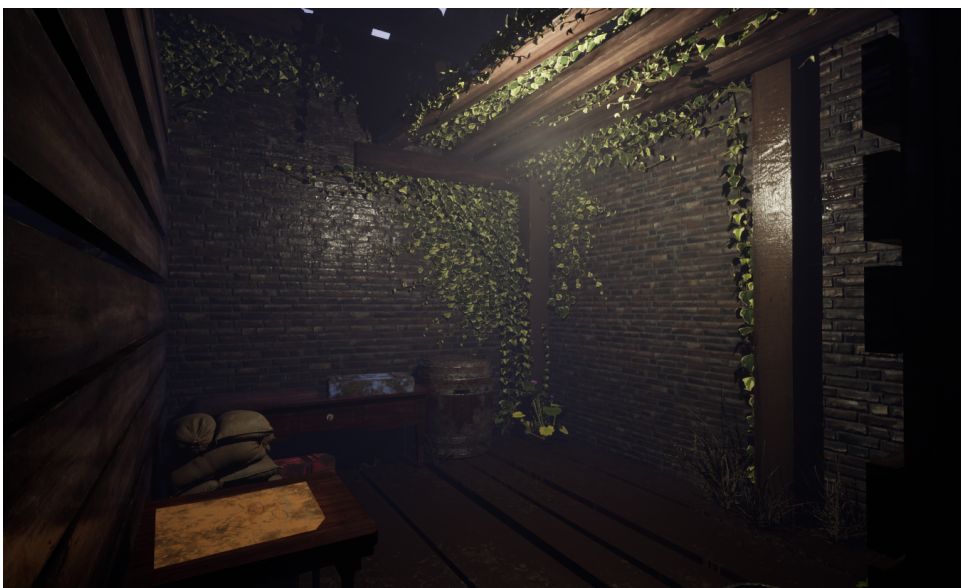
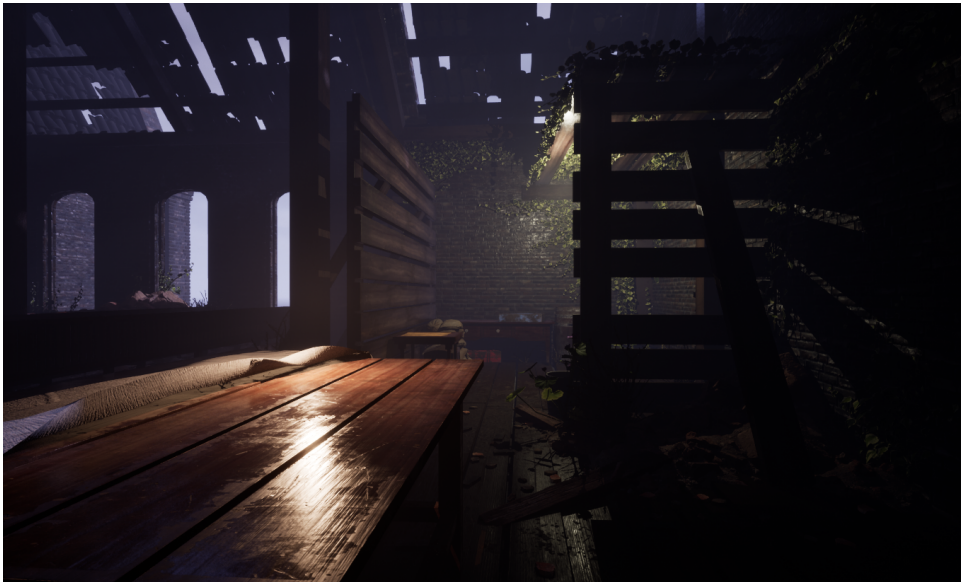
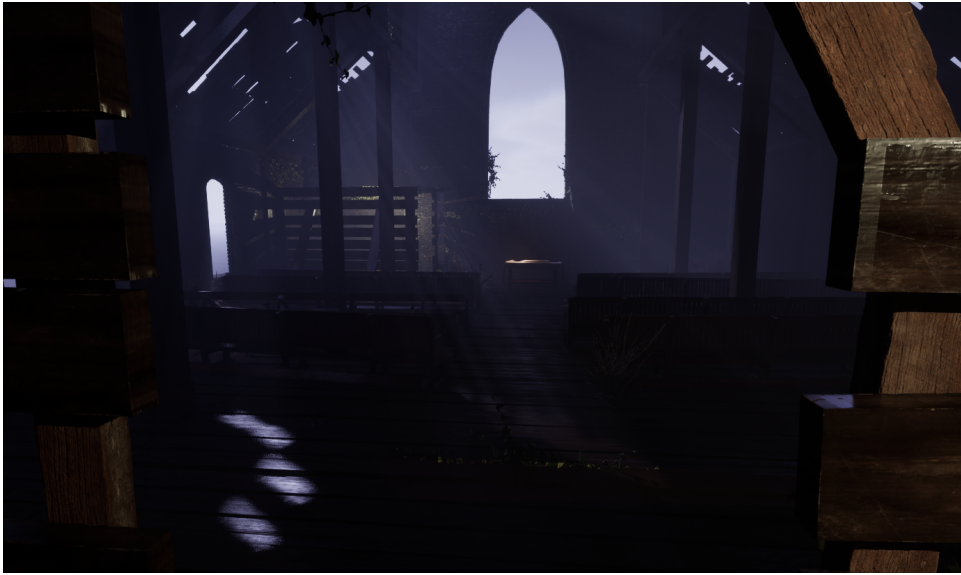
Una vez definida la silueta del proyecto, lo primero que se comenzó a desarrollar fue la creación de la iglesia. Tras exportar el mannequin de Unreal, pues sería la medida que se utilizaría, se comenzó con la creación de las piezas de la iglesia. Una vez se elaboró el primer paquete de piezas y tras crear las UV correctamente atendiendo a principios como el texel density, se comenzó con la creación de los materiales. Para esta parte, se usó el programa Mixer, que consta con una amplia galería de texturas escaneadas. Este programa permite crear los mapas de textura con textura de mezclado por altura y herramientas procedurales. Al término, se montó en Blender para una primera revisión. Paralelamente, se buscaba referentes sobre los distintos objetos que amueblaron la iglesia, así como seguir construyendo el entorno según la narrativa inicial donde este espacio, sería un lugar seguro para el jugador, un refugio. Se creó el proyecto en Unreal Engine, donde se iniciaron distintas

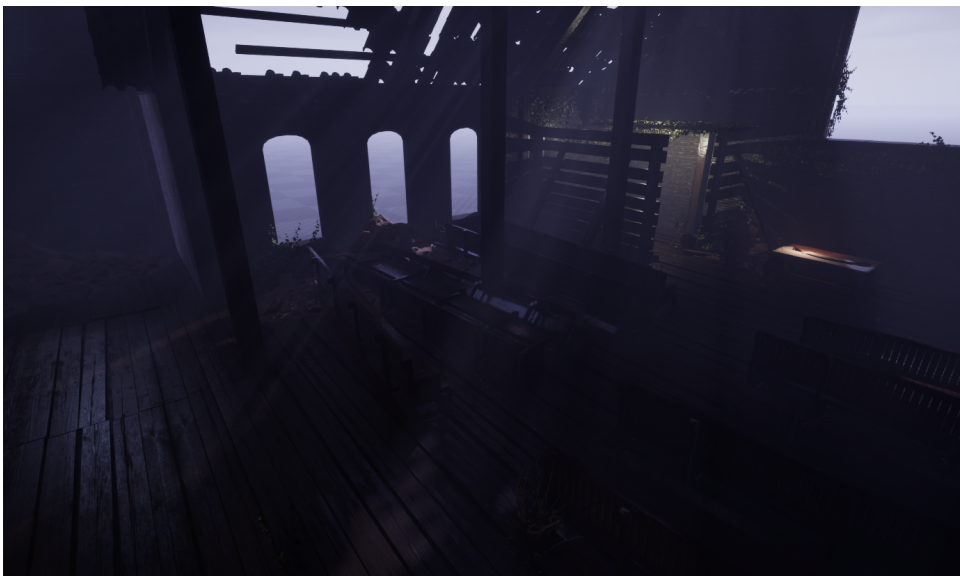
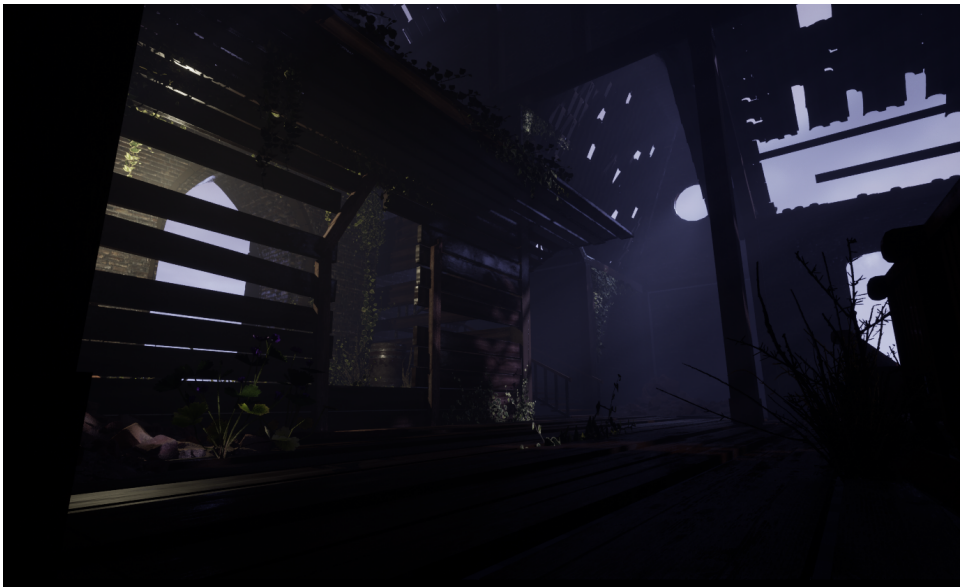
pruebas (para conocer mejor las novedades, que usar y qué desechar) y se realizaron las configuraciones iniciales. Tras importar todos los recursos creados, se inició el montaje en el propio motor.



Una vez montado todo con los props terminados (sólo los objetos más importantes que ayudan a visualizar mejor el espacio), se procedió a trabajar en la iluminación para generar un estilo personalizado y que satisfaga las ideas previas establecidas:







Este primer concept de la escena sirvió para poder visualizar mejor la estética, afinar la búsqueda de referentes... Paralelamente se buscaban recursos e información para temas de diseño como elaborar lluvia, movimiento del personaje, IA de los zombies...

Tras estas fases, se procedió a trabajar el terreno dentro de Unreal, con las herramientas para esculpir y pintar el terreno; se crearon los materiales necesarios y se comenzó a trabajar en Speedtree para crear la vegetación. Paralelamente, se comenzó el block out del poblado y la búsqueda de referentes arquitectónicos y planificación para elaborar las piezas modulares. Se montó el camino que conecta la iglesia con el poblado y se empezaron a implementar efectos como la lluvia, sonidos etc.



Tras la creación de las piezas del poblado, creando las texturas nuevamente en Mixer y Substance, se montó una versión del poblado para visualizar todo en conjunto, si se disimulaba la repetición, qué elementos decorativos añadir etc. Se crearon los puntos de conexión entre el camino y el poblado.



En este punto, se empieza a avanzar más en cuanto a optimización dentro del motor. A su vez, se comienza a ajustar todo el post procesado para potenciar la atmósfera y generar una fotografía singular.



En este punto relativamente avanzado, se decide dar un vuelco y rehacer el mapa, puesto que se necesitaba potenciar la idea de que la iglesia era un lugar seguro (Lo que requería estar más apartado) y a parte, se planteaba crear un mapa de mayor tamaño y con un perfil menos llano. Es de esta manera con la que se empieza a trabajar en un rediseño del mapa, pero únicamente del Landscape; destacar que construcciones como la iglesia, se juntan en un solo actor a modo de prefab para evitar tener que ser montado de nuevo (más a parte todas los beneficios de rendimiento). De aquí sale el terreno y el mapa definitivo:









Paralelamente, se empezó con el desarrollo del personaje; se elaboró primero una malla base atendiendo a prioridades como una topología lo más limpia posible y un flow de esta adecuado. De forma simultánea, se comenzó la realización del busto en Metahuman Creator y se recopiló

información sobre el sistema de movimiento que iba a usar el personaje (hasta ahora se estaba usando el mannequin por defecto).

En el personaje, tras modelarse, se procedió a crear la indumentaria en Marvelous designer; inicialmente se iba a seguir como referencia varias fotos recopiladas, pero se dio con el documento finalmente utilizado, una guía de patronaje escrita en 1887 y que reunía los requisitos estéticos que se buscaban, a aparte de dotar de mayor rigor y riqueza histórica al proyecto. Tras la creación de la ropa, se importó a Blender donde se limpió la malla, se hizo un remesh y se crearon las UV, atendiendo nuevamente, a los criterios de texel density y optimización. Se llevó a Substance Painter, donde se procedió a texturizar la indumentaria siguiendo los materiales reales de la época (Lana, algodón, cuero...) y sus patrones de estilo. Paralelamente se crearon los diversos detalles como el collar con crucifijo, los botones, el sombrero (que posteriormente se reemplazó por una boina, buscando un toque más inglés mediante el cliché). Se creó el rig del personaje, creando un esqueleto idéntico al que usa Advance Locomotion System, el sistema de movimiento del personaje (de esta manera, se evita retarget de todas las animaciones y la pérdida de virtual bones). Se unió el busto creado en Metahuman. Tras la implementación del personaje en el nuevo mapa, se comenzó a recuperar la estética de la iluminación y buscar referentes para la creación de la fábrica así como su punto de unión con el pueblo. Se desechó la idea de crear unos muelles, debido a que la extensión ya era notable y faltaba tiempo. Se comenzó con la creación de la fábrica, primeramente con su Block Out y finalmente la elaboración de sus piezas junto con las texturas; se crearon varios trim sheets. Destacar que en este punto, se cambiaron los árboles y parte de la vegetación. Se dio comienzo a la creación de las armas, modelado, UV y texturizado. Con la finalización se comenzó con la creación de los zombies. Mencionar que aquí salió la versión definitiva de Unreal Engine (anteriormente era un Early Access), lo que supuso una mejoría de rendimiento leve, puesto que se pulieron muchas funcionalidades.

La creación de los zombies fue similar a la del personaje, exceptuando el busto, que fue esculpido en Zbrush. Los elementos de ropa, nuevamente fueron hechos en Marvelous.

Una vez obtenido el rig y esqueleto de los zombies, se programó la IA de estos. Al término, se abordaron los retoques finales como la integración de sonidos, physical materials etc.

Por último, se procedió a la creación del menú principal, la programación de las funciones, creación de los sonidos y la banda sonora.

Desde este punto hasta la fecha de empaquetar el proyecto, se realizaron cambios en la estética tales como añadir un terreno, charcos de agua, pájaros, sonidos, efectos...

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. 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; esta se basa en tres partes diferenciadas:

Introducción - Donde centra al espectador en la acción, el tiempo y el lugar; presenta a los personajes, su situación y el porqué del inicio de su epopeya.

Nudo - Donde se desarrollan los acontecimientos desde la normalidad hasta la alteración en la historia del protagonista. A lo largo de la historia principal, han de resolverse otros contratiempos.

Desenlace - Termina los acontecimientos que sufre el protagonista.

En cuanto a estructura, también sería 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*.

Partiendo del esquema anterior, a nivel de desarrollo de personaje, este pasa por distintas etapas:

1.- **El mundo ordinario**, donde se plantean las cosas antes del conflicto. Se presenta al personaje en su día a día, de forma que cuando este se enfrente a su conflicto, el espectador conocerá qué es lo que abandona. En Videojuegos como Red Dead Redemption 2, el juego inicia saltándose este punto; te presenta a la banda huyendo de los agentes de policía, sin mostrarte antes esta parte. Sin embargo, conforme van sucediendo sucesos (mayoritariamente desgracias y complicaciones que ponen al protagonista en situaciones donde cuestiona su lealtad a la banda, sus creencias etc), van narrando situaciones del pasado que añora (en este caso, el punto sobre el que pivota fue un matrimonio fracasado).

Es decir, no es en la introducción de la historia y del personaje cuando muestran ese mundo ordinario, sino en la resolución de conflictos, en el final de la historia y la vida del personaje, cuando su historia acaba mal y está cerca de la muerte, cuando da al espectador y jugador la posibilidad (ya perdida) de un final e historia distinta.

2.- **La llamada a la aventura**, la cual se une a la parte de la introducción donde presenta el conflicto. A nivel de personaje, se le presenta un suceso disruptor o un objetivo no necesariamente de gran peso.

3.- **Rechazo de la llamada**, se puede considerar el *tira y afloja* del protagonista respecto a aceptar la aventura. Se debe de mostrar lo que el personaje abandona al salir de la zona de confort.

4.- **Encuentro con el maestro**, donde el personaje conoce a alguien que actúa como catalizador de la historia. Este personaje ayuda, da consejos y herramientas para que el héroe lleve a cabo la aventura. Volviendo al caso del título anterior, este elemento ya te viene desde el comienzo, siendo el

protagonista la mano derecha del líder de la banda (casi como un hijo). Este guía a la banda, aparentemente hacia un sueño y objetivo en común. En el caso de este título, el avance de la historia rompe con la relación haciéndolo enemigo y dando un vuelco a los acontecimientos. Nuevamente, es una línea de narración recurrente.

5.- **Cruce con el primer umbral**, donde el personaje hace frente al primer obstáculo que le presenta la aventura. Esto suele suponer el inicio del Nudo, pues ya no hay vuelta atrás para el protagonista. Siguiendo con el desglose del Red Dead Redemption 2, esta fase ya la superan antes del espectador conocer al personaje. Es decir, la presentación de este título, la forma en la que te presenta la situación, es omitiendo inicialmente los puntos anteriores y mostrándote una banda de forajidos huyendo de la justicia, donde tienen un líder carismático que actúa como padre del protagonista y se encuentra inmersos en una huida derivada de un plan fallido de atracar un banco (todo esto se va contando en las primeras horas de juego).

Aunque pueda parecer una forma brusca de adentrar al espectador, sin embargo, de realizarse correctamente es una forma idónea y plausible para Vindicator.

En el caso del juego de Rockstar Games, aprovecha el momento de confusión y abatimiento dentro del mundo que pueden tener los distintos personajes para que entre el jugador (o espectador). El jugador entra en el mundo sin saber que pasa, estando confundido a la vez que, los distintos personajes, han fracasado en un golpe, no saben cómo ha podido pasar, viéndose obligados a huir repentinamente del territorio, sin saber durante cuánto tiempo, si podrán volver a sus anteriores vidas... Es decir, se iguala tanto al jugador como a los personajes en cuanto a sensaciones e información.

6.- **Pruebas, aliados y enemigos**, esto es, durante el desarrollo del Nudo, los distintos conflictos a los que se enfrenta el protagonista, la presentación de un villano, aparición de aliados...

7.- **Acercamiento**, pues la superación de obstáculos (o resultado de estos) van conduciendo al protagonista a un reto definitivo. El protagonista ha evolucionado, conoce mejor a sus enemigos y a sí mismo. Volviendo al título usado como ejemplo, de la trama principal basada en escapar de la justicia y huir, se desarrollan conflictos relacionados por ejemplo con el territorio (Una zona donde huyen que desde hace generaciones, existen dos familias enfrentadas entre ellas y las cuales controlan el territorio y cómo esta banda se entromete), relacionadas con gente interna de la banda (uno de los integrantes, prestamista con pocos escrúpulos, se dedica a dar créditos con altos intereses a gente necesitada. Coloca al jugador en la tesitura de tener que cobrar el préstamo, viendo la minería y siendo posteriormente, un punto clave en la evolución del protagonista), relacionados con sucesos que han sido causa de otras decisiones anteriores (dentro del ruido de percances, aparece una historia amorosa entre el protagonista y la que fue en un pasado su mujer. Los problemas de la banda impiden al protagonista vivir este amorío por su estilo de vida criminal y fugitivo), sucesos inesperados (la aparición de tuberculosis en el protagonista, dando una fecha límite de vida) o incluso sucesos que abren la puerta a más conflictos (el secuestro del hijo de uno de la banda que obliga a esta a desplazarse a una zona peligrosa y enfrentar a varios villanos que atendiendo al problema principal, no hubieran salido a la luz).

8.- **Prueba suprema**, o el clímax de la historia e inicio del desenlace, donde el protagonista enfrenta el problema principal y cierra gran parte de las líneas abiertas anteriormente.

9.- **Recompensa**, después del reto decisivo. Esto es, lo que gana el protagonista tras completar su epopeya. Volviendo a la historia de ejemplo, en este caso, la banda se ha dividido en dos partes debido a un traidor. El protagonista muere, dejando un legado a otro miembro de la banda, convirtiéndose el protagonista que fallece, en el maestro de otro. Mencionar que la particularidad del título reside precisamente en esto, en la redención que únicamente es alcanzada tras una muerte trágica y llena de desgracias, donde lo único que has visto, es cómo el protagonista abandonaba sin saberlo su propia salvación, cayendo en un agujero cada vez más hondo y del que, sólo el tiempo y la memoria de sus amigos, lo recordarán como un héroe que dio su vida por ese perdón.

10.- **Camino de vuelta**, después de la prueba suprema, el protagonista debe de defender por última vez su recompensa.

11.- **Resurrección**, donde esa misma lucha a vida o muerte (puede ser en sentido figurado) muere el viejo yo del protagonista y nace el héroe purificado.

12.- **Regreso**, o la vuelta del protagonista al mundo ordinario esta vez, con la recompensa.

De esta forma, siguiendo la estructura propuesta anteriormente, se puede elaborar la trama de Vindicator de una forma cohesionada y con precedentes de su éxito. Faltaría la parte de confeccionar la historia en base a la época, el lugar, los acontecimientos y el pasado o presente del protagonista.

En este caso, el protagonista, de vuelta desde el nuevo mundo (EEUU) vuelve a su tierra natal, Inglaterra, dónde debido a una carrera por el desarrollo y la expansión industrial (1917), se han sucedido brotes de una enfermedad contagiada en las profundidades de cuevas mineras, donde las mismas empresas encargadas han desatendido el problema. Aquí en esta parte de la historia, se puede empezar a desarrollar tramas secundarias:

- Se fue buscando un cambio debido a una situación trágica (Muerte de su mujer e hijo en un parto, situación común en la época).
- Volvió debido a problemas en el territorio, lo que daría paso a conflictos futuros con la llegada de los que ha perjudicado, esta vez no como víctimas sino como villanos.
- Durante su ida/llegada, se ha visto enfrentado a conflictos (lo cuales no son mostrados al espectador sino contados) que serán resueltos o conllevarán a problemas/ventajas futuras.
- El personaje es bien recibido a su vuelta o no debido a situaciones pasadas.
- Se presenta a su vuelta la situación de esta nueva enfermedad. El protagonista se da cuenta rápido y es de los pocos en hacerlo con lo que no le creen; no se da cuenta hasta que es demasiado tarde; es consciente pero le resta importancia o está sobornado por la empresa...
- Diversas cuestiones relacionadas con su figura religiosa: A partir de una situación desencadenante, desde siempre, por poder...
- Relacionadas con cómo se presenta al espectador: Un hombre con una moralidad intachable dispuesto a luchar por el bien; con unos principios cuestionables pero que sabe distinguir entre lo que está bien y lo que está mal, actuando muchas veces erróneamente pero con consciencia sobre esto; actúa sin conocer las consecuencias de sus actos, tanto si perjudica al prójimo como si lo ayuda, solo busca su bien.

- Sobre la empresa, qué relación tiene con el protagonista, si es ajena a este o si han cruzado alguno de sus miembros vivencias con este. Mayor desarrollo de sus integrantes, motivaciones, miedos (más allá de la búsqueda de fortuna o de perder sus puestos de trabajo si desobedecen órdenes) o si por el contrario, la empresa se presenta como un villano sin definir (similar a Abstergo en la saga de juegos de Assassins Creed, donde presentan sólo a villanos y altos cargos los que vencer)...
- Sobre la situación desencadenante del núcleo principal (Expansión del virus, los zombies toman las calles, caos y descontrol absoluto, caída de gobiernos...); se encuentra el personaje sólo ó cuenta con amigos/aliados. De estar solo, ¿Desarrolla apego por algún compañero, ya sea animal o inanimado (Similar a la película *Náufrago*, donde el actor Tom Hanks protagoniza a un náufrago que encuentra en Wilson, una pelota, un amigo al que le dota de cualidades humanas)?, ¿Encontrará aliados a lo largo de la historia? O está condenado a vagar solo como moraleja a una acción pasada (abandono de su familia, desinterés en un hijo bastardo...).

E infinidad de cuestiones que pueden seguir añadiendo desde un comienzo mucho lore e historia al personaje.

Lo mismo sucede a la hora de desarrollar tanto la trama o nudo principal, como el desarrollo del personaje desde la llamada a la aventura (ya presentada anteriormente) hasta la prueba suprema:

- Relacionados con el mundo, ¿Qué desarrollo ha habido fuera del protagonista? ¿Se presentan situaciones similares a *The Walking Dead*, donde la mayor parte de contratiempos sucede debido a los supervivientes que, enloquecidos y al ver que la civilización y moral ha muerto, se pretenden regir como dueños y soberanos del nuevo mundo? Ó se presenta un mundo dividido en pequeñas tribus cada vez más debilitadas por el avance de los zombies, quienes se presentan como un actor difícil de superar (Similar a películas como Guerra Mundial Z, donde los zombies arrasan al ser humano).
- Relacionadas con la llamada a la aventura para el héroe, ¿Se ve motivada por la posibilidad de una cura?, ¿Motivado por la supervivencia?, ¿Por encontrar a alguien del pasado, un amor, una hija/hijo del que abrir una nueva línea argumental?.
- Sobre la figura del maestro, ¿Encuentra alguien que sepa más sobre esta enfermedad?, ¿Encuentra al líder de un grupo de supervivientes que le ayuda en algún punto?, ¿Situación similar a la de *The last of Us*, donde es el protagonista el mentor o protector de alguien, como su hija/hijo o una figura que represente un ser inocente e indefenso, alguien conocido en su pasado...?
- Sobre la información que va adquiriendo, cómo se le descubre los causantes, posibles curas, nuevos conflictos (por ejemplo, entre grupos de supervivientes, milicias a sueldo del gobierno queriendo derribar cualquier atisbo de vida en la zona, sea zombie o no; los zombies como sujeto más inteligente, se agrupa en grupos jerarquizados y comienzan ataques más organizados, similar a *Soy leyenda*; la propia empresa minera queriendo encubrir cualquier investigación, traiciones dentro de la propia banda de supervivientes...). Aquí entran las Pruebas, aliados y enemigos, como se ha ejemplificado anteriormente. También se pueden rescatar tramas abiertas al comienzo sobre su viaje y/o vuelta de EEUU, líneas no contadas hasta ahora como el pasado del protagonista, que aparece ahora esa información en forma de conflicto, enfermedad...

Y finalmente el clímax de la historia, el cual no se esclarecerá a continuación debido a que es la respuesta a todas las posibles tramas abiertas anteriormente y depende por lo tanto, del desarrollo de estas (sin atender a cuestiones relacionadas con dejar cierto hilo argumentativo para segundas partes).