UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCUELA POLITÉCNICA SUPERIOR DE GANDÍA

GRADO EN COMUNICACIÓN AUDIOVISUAL

UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

"Producción de un escenario óptimo para videojuegos 3d"

TRABAJO FINAL DE GRADO

Autor/a:

Malena Galán Puerto

Tutor/a:

Jose Antonio Lozano Quilis

***GANDIA**, 2021*

RESUMEN

En este Proyecto de Final de Grado se propone llevar a cabo la realización de un escenario optimizado para videojuegos 3D basado en el concurso ofrecido por la web de The Rookies: *"World of Real-time"*. En él se explican los procesos y herramientas utilizadas para llevar a cabo las diferentes disciplinas que envuelven el proyecto.

Abordaremos cada una de las fases de desarrollo partiendo de dos *Concepts Art* del artista Svetlin Velinov hasta realizar una cinemática donde se muestre el resultado final. Entre estas fases se encuentran además de la escultura digital, etapas como el texturizado, la creación de vegetación, la realización de *shaders*, la iluminación, incluyendo el renderizado del mismo.

La finalidad de este proyecto es obtener un producto profesional donde poder mostrar los conocimientos y capacidades adquiridos y utilizarlo como base de porfolio para la futura salida laboral.

Palabras clave:

Videojuego 3D, The Rookies, escenario, esculpido, modelado, texturizado, Zbrush, Blender, Unity, Photoshop, substance Painter, Marmoset, 3D.

ABSTRACT

In this final year project, it is proposed to carry out the creation of an optimized 3d game environment based on the contest offered by the The Rookies web: "*World of Real-Time*". There it is explained all the processes and tools needs to carry out the different disciplines that are involved in the project.

We will tackle each phase of the development starting with two concepts art from the artist Svetlin Velinov until the creation of a cinematic where the result will be shown. Among these parts, we can find digital sculpting in addition to stages like texturing, foliage creation, shader implementation, lighting, and the rendering itself.

The project's objective is to obtain a professional product to show all the knowledge and capabilities acquired and use them as a portfolio base to create nice job opportunities.

Keyword:

3D videogame, The Rookies, environment,sculpting, modeling, texturing, Zbrush, Blender, Unity, Substance Painter, Marmoset, 3D.

ÍNDICE

1. INTRODUCCIÓN

El presente trabajo práctico consiste en el diseño y la elaboración de un escenario de videojuego 3D y su implementación dentro de un motor de videojuegos. La creación de dicho escenario vino motivada por la web The Rookies quien abrió el concurso "*World of Real-time*", idóneo para adquirir capacidades contundentes que permitan desarrollar un perfil profesional de artista 3D orientado al sector de los videojuegos.

La realización de este proyecto se ha desarrollado siguiendo los diferentes flujos de trabajo utilizados en la industria que han permitido obtener la máxima optimización en cada uno de los objetos y así garantizar el funcionamiento y mayor rendimiento del escenario.

En lo que se refiere al trabajo escrito, el cuerpo está estructurado respondiendo a las necesidades constructivas y gráficas del proyecto siguiendo este orden: referencias, blocking y conceptualización, escultura digital, retopología, mapeado de UV, Bake, texturizado, vegetación, shaders, montaje, iluminación y postprocesado.

2. OBJETIVOS

Para la realización de este Trabajo Fin de Grado se proponen los siguientes objetivos:

Componer, elaborar e implementar un escenario para videojuego 3D en un motor de juegos, asumiendo el modelado 3d, texturizado y postprocesado, para conseguir una calidad gráfica profesional que permita reproducir adecuadamente la estética del proyecto y sea funcional en plataformas de medio-alto rendimiento.

Integrar y desarrollar los conocimientos adquiridos tanto durante el periodo de formación en el grado de Comunicación Audiovisual, así como del ciclo superior de Animación 3D, Juegos y Entornos Interactivos y los obtenidos durante la elaboración de este proyecto.

Seguir los requisitos que especifican las bases y los plazos establecidos del concurso que ofrece la Plataforma The Rookies.

3. THE ROOKIES

The Rookies[1] es una comunidad fundada en 2010 para artistas digitales no profesionales de la que forman parte varias instituciones de alto nivel por todo el mundo. Esta web ofrece a sus diferentes miembros una forma de mostrar su trabajo al mundo compartiendo su viaje creativo, logros y mejoras.

Fig. 1. Logo web The Rookies. Recuperado de: https://cutt.ly/TWvM4q5

Esta plataforma cuenta con diferentes aspectos que ayudan a los usuarios a desarrollar sus habilidades como un blog donde artistas comparten artículos interesantes sobre el sector, un canal de Discord para poder consultar y compartir sus progresos, o pequeños concursos/desafíos para impulsar su creatividad. Además, The Rookies concede con regularidad diferentes logros e insignias para motivar a sus miembros.

De este modo, la web recoge a jóvenes creativos en efectos visuales, animación, juegos, realidad virtual, gráficos en movimiento y visualización en 3D y los divide en diferentes niveles con el fin de acompañarlos e impulsarlos a lograr sus objetivos dividiéndolos de este modo:

-*Debut*. Se trata del inicio del aprendizaje donde se adquieren los conocimientos de software y herramientas y no posee mucho trabajo para compartir.

-*Player*. En esta etapa desarrolla nuevas habilidades y fortalece su confianza de la mano de un centro educativo o entrenando en su tiempo libre.

-*Contender*. Es la persona que ha desarrollado un portfolio listo para compartir con proyectos a nivel de la industria y estableciendo contactos para una oferta de trabajo.

-*Rookie*. Se trata del nivel máximo y se alcanza durante el primer año de empleo.

De este modo, las empresas pueden ver de forma rápida en qué nivel se encuentra el usuario además de su porfolio, concursos, logros, insignias, etc. (The rookies, 2020)

[1] Web The Rookies: https://www.therookies.co/

Concurso *"World of Real-time"*

La web The Rookies ofrece cada cierto tiempo una serie de pequeños desafíos/concursos para impulsar las habilidades de los diferentes miembros de la comunidad. Así pues, este Trabajo Final de Grado engloba la ejecución desarrollada para la participación en "*World of Real-time*" por lo que me dispongo a exponer las bases de dicho concurso.

Consiste en desarrollar un escenario abierto que incluya diferentes tipos de vegetación y recrearlo con un render dentro de un motor en tiempo real. Así pues, para ayudar a inspirar la creatividad ofrecieron el siguiente guion:

"En un planeta inexplorado durante muchos años, nuestros dos viajeros ahora están perdidos en este fabuloso lugar lleno de misterios.
"Hola Coronel, ¿está bien? Esa brecha temporal en realidad me hizo vomitar un poco ".
"Respire hondo Capitán y agárrese de esa bolsa de enfermo, porque no va a creer lo que estoy mirando por esta ventana. Honestamente, no tengo idea de cómo diablos vamos a salir de aquí ".

Respecto a los requisitos del concurso se exige que la escena debe estar renderizada haciendo uso de un motor en tiempo real como EU4, Unity, Eevee; debe evocar asombro e incluir vegetación, follaje y una fuerte presencia de narración que refleje el guion proporcionado. Además, se permite utilizar el arte conceptual de otro artista y darles crédito por su contribución y está permitido el uso de herramientas como Speedtree, Megascans[2] entre otras. (The Rookies, 2021)

Por otro lado, el concurso ofrece, además del guion proporcionado, una serie de imágenes de referencia (ver ver Fig. 2) para guiar al artista a orientarse en la ambientación de la escena.

[2] Megascans: Galería de assets generados a través de fotogrametría.

Fig. 2. Imágenes de referencia proporcionadas por el concurso de The Rookies.
Collage propio.

4. METODOLOGÍA

Partiendo de las referencias visuales y especificaciones proporcionadas por el concurso "*World of Real-time"* (desarrollar un mundo abierto que evoque asombro, incredulidad y emociones), se seleccionaron dos concept arts del artista Svetlin Velinov que mantienen una estética y atmósfera similar.

A continuación, se procedió al desarrollo gráfico dividido en seis procesos fundamentales:

El *blocking* del escenario, que se realizó principalmente con 3ds Max y Blender, partiendo de un estudio previo de los *concept art* para integrar y posicionar los objetos en el escenario. Se crearon diferentes modelos con un nivel bajo de polígonos para ajustar el tamaño y posicionamiento de los objetos.

El *esculpido digital*, realizado en Zbrush, donde se han generado los modelos de alta poligonización ajustando la volumetría y proporciones de los modelos utilizados en el *blocking* y se ha añadido el nivel de detalle deseado.

La *retopología*, realizada en 3DS Max, partiendo de los modelos con alta calidad en cuanto a detalle se refiere, se reorganizó la topología con el fin de disminuir el número de polígonos que componen los assets para garantizar el funcionamiento del escenario en el motor de juegos.

El *mapeado de uvs* está realizado en el mismo software utilizado en la fase anterior y comienza por el *unwrapping* o despliegue de coordenadas *Uv*, donde se deciden las costuras (*seams*) para el despliegue de la malla poligonal. Una vez comprobado que el *unwrapping* es correcto, se plasmaron los detalles de la escultura digital en los modelos optimizados mediante el proceso de *bake* haciendo uso del software Marmoset. Este proceso genera diferentes mapas como el de normales, difuso, oclusión, metálico, etc.

El *texturizado* se llevó a cabo mediante Photoshop y Substance Painter haciendo uso de los mapas generados en el *bake*. Por otro lado, la vegetación se desarrolló en Photoshop añadiendo un canal alfa para, posteriormente, estructurar en Blender la forma necesaria.

A continuación, se llevó a cabo el montaje de todos los assets con sus respectivas texturas dentro del motor de juegos Unity y se incluyó el shader de agua, la iluminación y las partículas.

La última fase del proyecto fue el postprocesado y la creación de los renders y la cinemática final del escenario.

Así pues, en el gráfico siguiente podemos ver un esquema del flujo de trabajo:

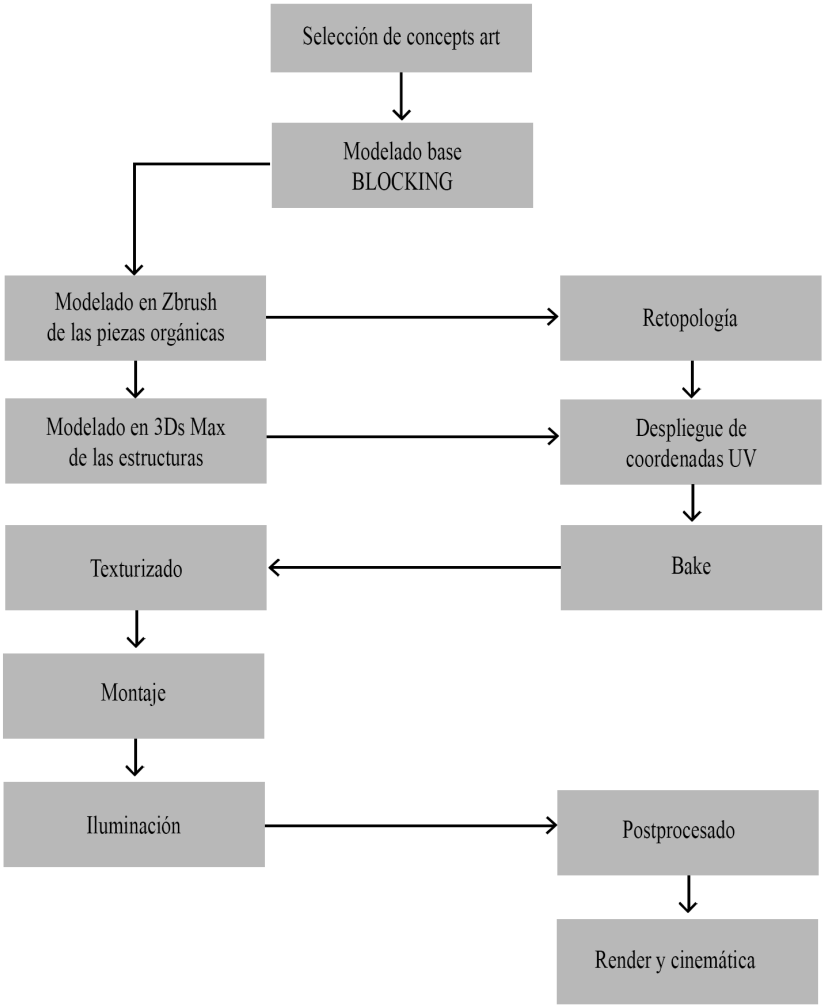

Fig. 3. Proceso de trabajo del proyecto. Elaboración propia.

5. TÉCNICAS Y PROGRAMAS

5.1 TÉCNICAS

5.1.1 MODELADO

Antes de empezar a definir los tipos de modelado, cabe recalcar que en una producción no solamente se utiliza un único tipo de modelado, de hecho, combinándolos conseguimos mejores resultados. Esto nos proporciona un abanico de posibilidades mucho mayor.

1) Box Modeling

El *box modeling* suele ser el que más se utiliza, además de que es el primero que se enseña en cualquier formación de 3D. Este modelado consiste en que, mediante la manipulación de caras, vértices y aristas del modelo, podemos ir extruyendo[3] y creando nuevas caras, para ir dando forma al objeto que queremos conseguir. Este modelado suele partir de formas primitivas como cubos, planos, cilindros, esferas… para conseguir la mayor parte de las formas. A partir de estas primitivas, junto con herramientas básicas como la creación de *loops, bevels* y el ya nombrado *extrude*, podremos manipular las formas y conseguir el resultado deseado. Este tipo de modelado se suele relacionar mucho con el modelado *hard-surface* (objetos arquitecturales u objetos hechos por el hombre y máquinas). También se puede utilizar para el modelado orgánico, sin embargo, hay técnicas que facilitan su realización.

Si nuestra intención es crear una versión *high poly*[4] con este método, lo que se hace es poner un modificador de subdivisión a nuestro objeto, para así suavizar y redondear la superficie. Lo que realmente hace en un proceso interno donde se subdivide la geometría mediante un algoritmo llamado *catmull-clark*. (Selin, E. 2021).

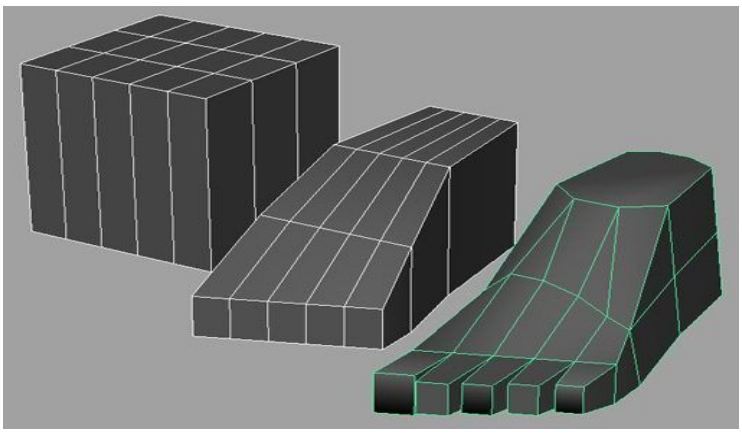

Fig. 4. Modelado mediante la técnica *box modeling*.
Recuperado de: https://cutt.ly/nWv0Oka

[3] Extruir: Dar forma a la malla haciéndola salir o entrar por una obertura anteriormente dispuesta
[4] High poly: modelo 3D con alto número de polígonos

2) Polygon Modeling

El modelado poligonal y el *box modeling* vienen a ser muy similares y se complementan todo el tiempo durante el proceso de creación de un *asset*[5]. La diferencia principal es que, en este modelado no empezamos con primitivas, sino que empezamos con vértices y aristas haciendo todo el borde del objeto sin ningún tipo de profundidad. Una vez tengamos la silueta será entonces cuando añadamos la profundidad (ver Fig. 5).

Esta técnica se suele llevar a cabo con formas bastante más complejas (como por ejemplo ornamentos) y también si poseemos la vista frontal y de perfil de nuestro objeto. De esta forma conseguiremos una precisión bastante alta.

Las herramientas tanto para el modelado poligonal como el *box modeling* son las mismas herramientas, sin embargo, se utilizan de forma diferente. (Selin, E. 2021).

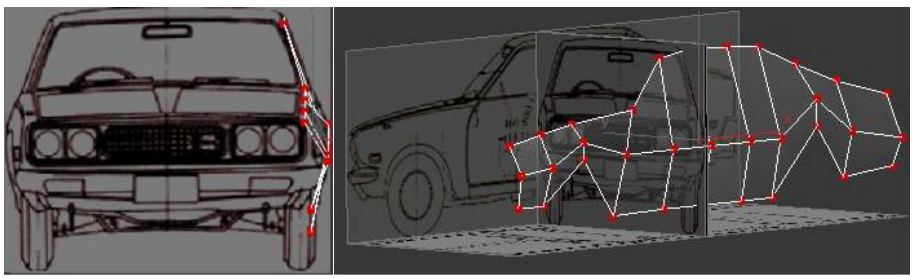

Fig. 5. Modelado mediante la técnica *Polygon Modeling*. Collage propio.

3) NURBS and curve

En primer lugar, las siglas corresponden a "*non-uniform rational b-spline*" o *B-splines* racionales no uniformes. Este método es completamente diferente a los anteriores, ya que en este creamos las formas mediante curvas que se gestionan mediante puntos de control (ver Fig. 6). Podemos interpretar el espacio que existe entre dos curvas para crear puentes entre ellas.

[5] Asset: cada uno de los elementos que forman parte de un escenario 3d.

Este modelado se utiliza mayoritariamente en ingenierías y en CAD. Suele ser un modelado mucho más preciso y matemático y no tan artístico.

El hecho de modelar con *nurbs* es muy similar a los vectores en 2D. Cuando escalamos una imagen que tiene poca resolución se verán los píxeles.

Del mismo modo, si escalamos un objeto modelado poligonal o nos acercamos se podrán ver esas caras más grandes. Sin embargo, cuando hacemos grande un icono vectorial, este no pierde resolución, ocurre lo mismo con este tipo de modelado. (Selin, E. 2021).

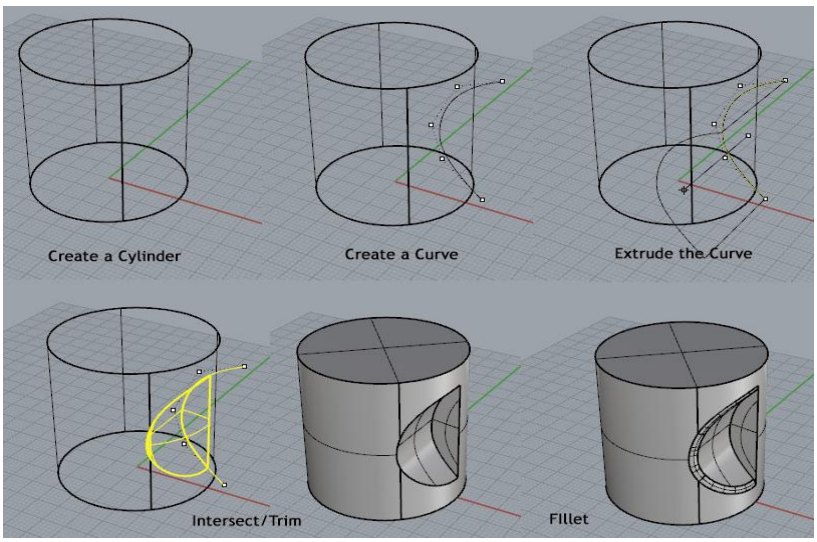

Fig. 6. Modelado mediante la técnica *Nurbs and Curve.*
Recuperado de: https://cutt.ly/jWv0bSe

4) Escultura digital

La escultura digital nos aleja del lado más técnico del 3D y nos lleva al lado más artístico. Con esta técnica podemos manipular la geometría como si se tratase de bloques de arcilla mediante unos pinceles que nos permiten modificarla. Aquí la cantidad de geometría no es un problema ya que lo estaremos utilizando para crear modelos *high poly* con una muy buena resolución (ver Fig. 7). Sin embargo, esto implica que, con tantísima geometría, tras el proceso de esculpido se tengan que

realizar algunos ajustes para reducir la malla y así ser óptima. Aun así, es un muy buen recurso para crear ilustraciones, ya que ahí lo que realmente importa es su visualización y no tanto la optimización o el orden de la geometría.

Fig. 7. Personaje esculpido de Overwatch en Zbrush.
Recuperado de: https://cutt.ly/uWv09yj

En el sector de los videojuegos y de cine cada vez se está popularizando más la escultura digital para la creación de personajes, animales o criaturas. Además, muchos *concept artists*[6] utilizan la escultura digital para crear una base en un momento de aquello que quieren conceptualizar y simplemente con una captura de pantalla, pueden empezar a trabajar sobre esa escultura. Este es un *workflow*[7] bastante rápido e iterativo. (Selin, E. 2021).

5) Otras técnicas:

Tras explicar las 4 técnicas más comunes para la realización de modelos 3D, explicaré algunas que también juegan un papel importante en la producción. Estas son:

Fotogrametría: consiste en realizar un montón de fotos a un objeto de la vida real para literalmente escanearlo y poder meter el objeto en el ordenador. Mediante una serie de cálculos el programa combina todas las fotos hechas creando de ahí un modelo en 3D.

***Procedural modeling*:** últimamente se está hablando mucho de este tipo de modelado, ya que posiblemente sea capaz de suplantar muchos *workflows*. Por

[6] Concept artists: persona que genera el desarrollo visual de una producción.
[7] Workflow: Flujo de trabajo

ejemplo, alguien puede programar un plugin que genere edificios. En ese plugin mediante sliders o parámetros somos capaces de decidir los pisos que tiene, las ventanas, si posee escalera de incendios o no. Por esta razón, este tipo de modelado es una buena apuesta debido a que podemos generar una gran variedad de objetos en un momento. Si tuviésemos que modelar cada uno de los edificios del escenario de forma única, en términos de costes de producción serían altísimos.

***Boolean modeling*:** El modelado por booleanas es un tipo de modelado que se suele complementar mucho tanto con el *box modeling* como con el modelado poligonal. Consiste en 3 operaciones básicas: unión, intersección y diferencia.

Según la imagen expuesta (ver Fig. 8), consiste en realizar estas operaciones a dos geometrías, para conseguir formas mucho más complejas que mediante el modelado tradicional tendría una complejidad alta. (Selin, E. 2021).

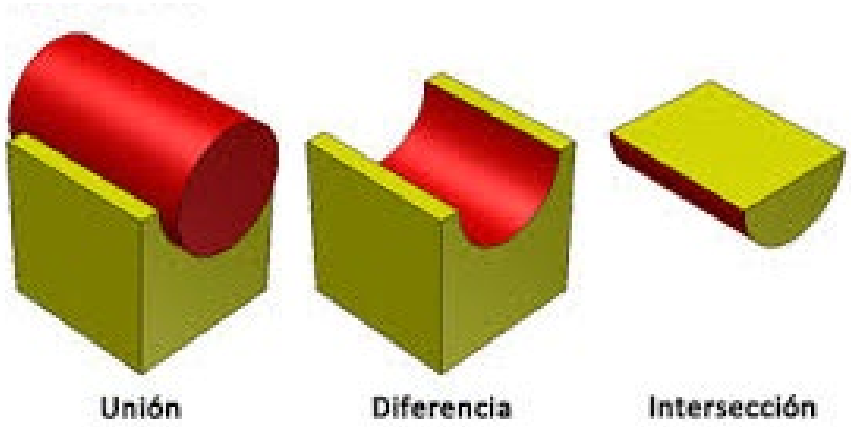

Fig. 8. Modelado mediante la técnica Booleans.
Recuperado de: https://cutt.ly/yWv2FeX

5.1.2 RETOPOLOGÍA

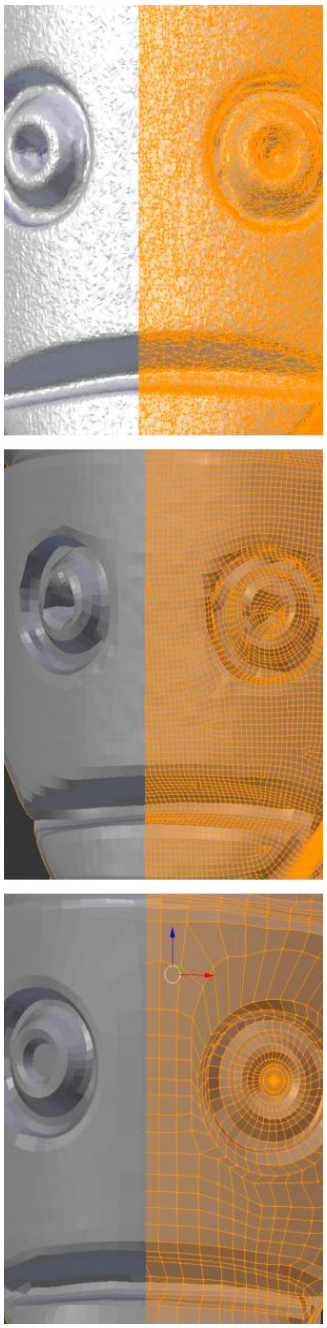

Fig. 9. Ejemplo de modelo *highpoly*, retopología automática y manual. Collage propio.

La retopología es un proceso mediante el cual construimos una malla simplificada a partir de otro modelo en 3D con más detalle. Se trata de construir cada polígono para que este se acople al modelo con más resolución y conseguir su forma. Este proceso es completamente necesario dentro de una producción 3D, siempre que se tenga un modelo de alta poligonización, ya que trabajar con un modelo tan pesado en un motor de videojuegos puede afectar drásticamente al rendimiento de este. Hay dos tipos de retopología: automática y manual.

La retopología automática pretende crear una versión rebajada de ese modelo *high poly* mediante algoritmos que consideran la localización de los polígonos para mantener la forma lo mejor posible. Es un proceso rápido y sencillo, sin embargo, es poco recomendable cuando el objeto que estamos haciendo tiene que ser animado. Esto es porque, los programas no suelen tener en cuenta los *loops* para que los personajes se puedan deformar de forma correcta, y si lo hacen aún no son lo suficientemente precisos para que quede perfecto.

La retopología manual, como dice la palabra, es la que construimos nosotros polígono a polígono. Con esto conseguimos tener un absoluto control de la topología y un resultado excelente. La mayor desventaja es que es un proceso muy lento y tedioso.

Los programas que más se utilizan para la realización de estos trabajos son: para una retopología automática es muy famoso el *Zremesher* de Zbrush o su *Decimate* y para la manual cualquier programa de modelado se podría utilizar, sin embargo, el más utilizado es Autodesk Maya (Ver Fig. 9). (Trazos, 2021).

5.1.3 *UVS*, BAKE Y TEXTURIZADO

Despliegue de *UV's*

En primer lugar, empezaremos con las *Uv's* ya que es un paso necesario para realizar los procesos posteriores. Las *Uv's* son una textura de coordenadas que recogen la información de los vértices de nuestra geometría. Las *Uv's* permiten crear un vínculo entre la malla 3D y cómo las texturas se van a aplicar sobre ellas. Básicamente son como unos puntos de control que marcan qué píxeles de la textura corresponden a qué vértice del objeto 3D. Para conseguir sacar estas *Uv's*, cogemos el modelo y vamos haciendo cortes para que se pueda desplegar correctamente. Para que se entienda de forma correcta, las *Uv's* funcionan como cuando montamos un cubo de papel (ver Fig. 10). (Pluralsight, 2017).

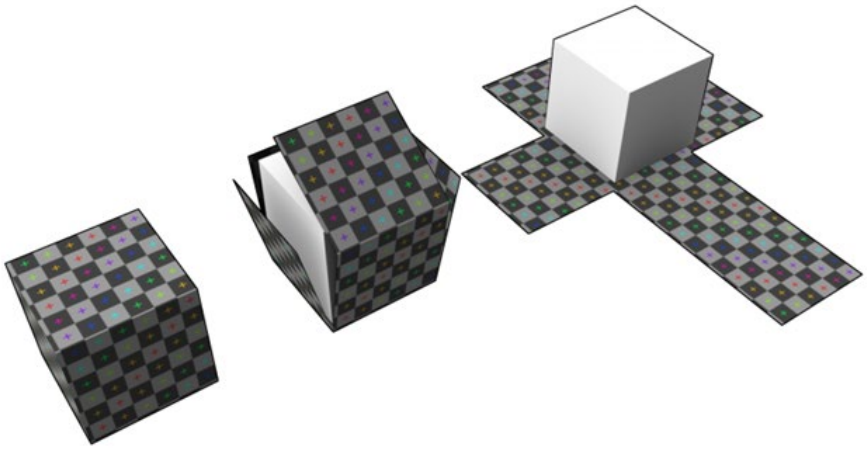

Fig. 10. Ejemplo de despliegue de *Uv's*.
Recuperado de: https://cutt.ly/WWv3CdG

Bake

Las *Uv's* son necesarias para el proceso del *bake*, sin ellas la información que saquemos no se podrá aplicar en ningún sitio. De este modo el *bake* se utiliza para poder proyectar sobre una geometría con un poligonaje bajo, una textura con los detalles del modelado de alta. Esto crea una ilusión que da a parecer que el detalle se encuentra puesto en el *low poly*, sin embargo, solamente es textura, y no

geometría. En el *bake* se pueden generar distintos mapas para proyectar esa información.

El mapa más importante es el *Normal Map* o mapa de relieves que, interactuando con la luz, es capaz de simular esos detalles en la superficie del modelo sin necesidad de más geometría.

Hay otros mapas que se utilizan en la fase de texturizado para obtener información del modelo 3D y facilitarnos así su pintado. Como, por ejemplo, el *Ambient Occlusion, el Curvature, el Thickness o el Position* (ver Fig. 11).

El *Ambient Occlusion* genera sombras de contacto entre los objetos proporcionando una información muy valiosa a la hora de texturizar.

El *Curvature* es capaz de calcular los bordes del modelo para poder añadir ralladuras y cortes, ya que estos suelen ser los más expuestos a los golpes.

El *Position* calcula el lugar en el que se encuentra el objeto en el espacio 3D. Conocer esto nos permite añadir gradientes[8] al objeto para crear humedades, variacones de color…

El *Thickness* es otro mapa que conoce la densidad de un objeto. También se puede utilizar para crear máscaras en el modelo con el fin de conseguir un efecto de SSS (Sub-Surface Scattering) que permite simular como la luz atraviesa los objetos.

[8] Gradiente: aumento o disminución progresiva del color que determina la profundidad y densidad de un objeto

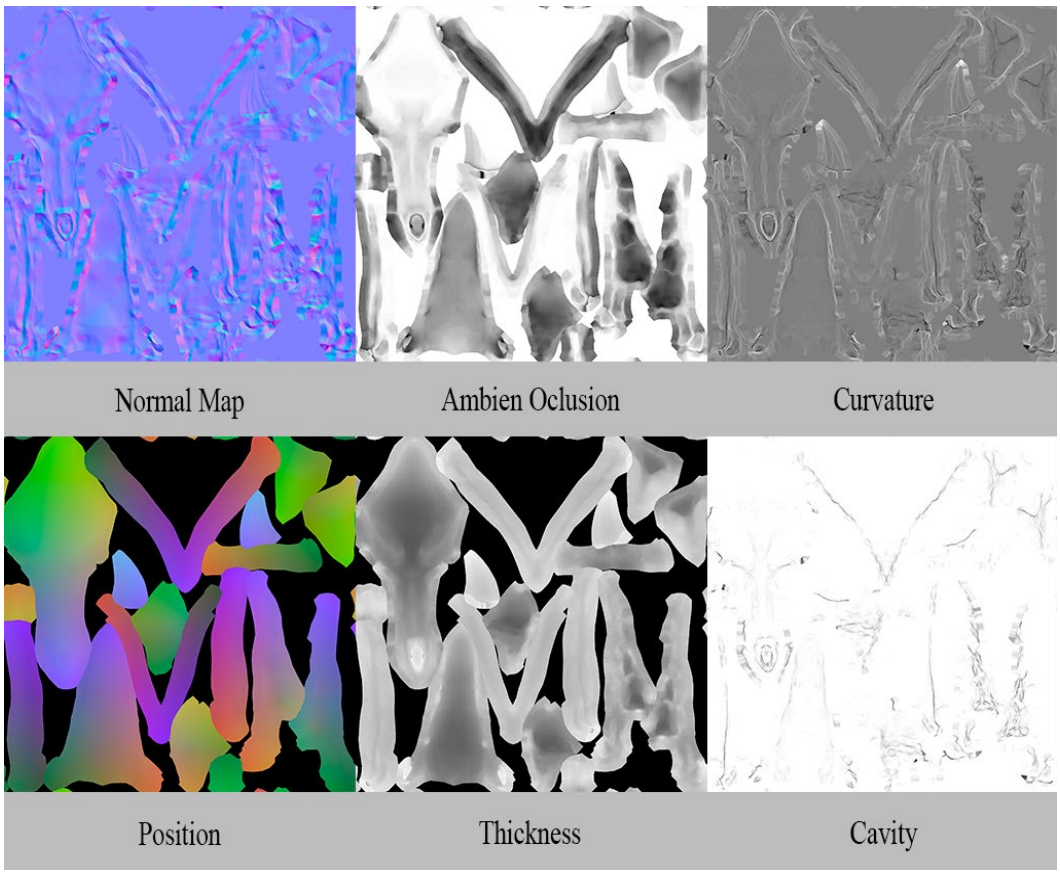

Fig. 11. Mapas de *Bake*. Elaboración propia.

Texturizado

Finalmente, el texturizado consiste en añadir textura a nuestro objeto ayudándose de los mapas citados anteriormente. En este artículo diferenciaremos dos tipos de texturizado, el texturizado *handpainted* y el texturizado PBR o *physical based rendering*.

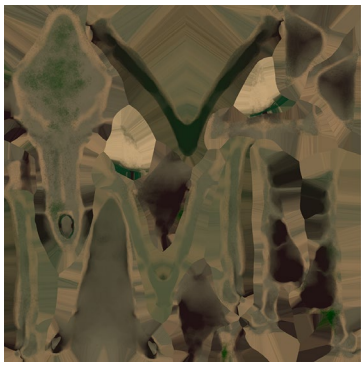

Fig. 12. Mapa Albedo del cráneo del proyecto. Elaboración propia.

El texturizado *handpainted* utiliza simplemente una textura albedo[9] (ver Fig. 12) para simular todos los efectos y detalles en el modelo. Estos objetos no interaccionan con la luz, y a la hora de ser pintados suelen tener una luz predefinida y

[9] Albedo: mapa que recoge el color de un modelo.

homogénea en todo el personaje que se añade a la textura (ver Fig. 13). (Bourykina, Y. 2018).

Fig. 13. Proceso de texturizado estilizado. Recuperado de: https://cutt.ly/sWv8SNC

Las texturas PBR o *physical based rendering* hacen referencia a una técnica de renderizado que permite calcular como reflejan las luces y las sombras que producen los objetos de una forma más realista. Las texturas PBR utilizan el mapa de normales, el color del material o base color y el *heightmap* (desplazamiento de los polígonos). Hay dos *workflows* para el PBR: el *specular-glossiness* o el *metallic-roughness*.

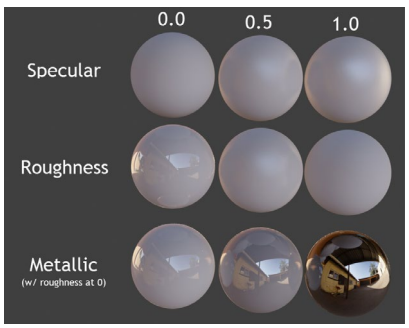

Fig. 14. Valores de texturizado Metallic-Roughness.
Recuperado de: https://cutt.ly/nWEeLFJ

El más común y el más sencillo de entender es el *metallic-roughness* ya que solamente cabe entender si el material que intentamos recrear es metálico o no y lo rugosa que es su superficie. Estos valores se mueven entre 0 y 1, siendo 0 un material dieléctrico y 1

un material totalmente metálico, y lo mismo para el *roughness*, 0 muy liso y 1 totalmente rugoso.

El *specular-glossines* no se utiliza tanto ya que para cada uno de los materiales hay un valor especular o de reflexión único, por eso es bastante más complicado conseguir el efecto deseado en este tipo de proceso (ver Fig. 14). (Texturas 3D, 2020)

En este proyecto he utilizado el texturizado PBR con *metallic roughness*.

5.2 PROGRAMAS UTILIZADOS

En una producción tanto de escenarios como personajes de videojuegos se utilizan una gran cantidad de programas para cada uno de los procesos. Aunque haya algunos programas que sean capaces de realizar varias tareas, cada uno de los programas se suelen especializar en una parte de los procesos.

-Modelado:

1) *Blender y 3Ds Max*: son dos programas que se utilizan principalmente para la parte de *box modeling* y *poly modeling*. Blender puede utilizarse también para la parte de escultura digital, sin embargo, como he comentado anteriormente hay algunos programas que se especializan mucho más en algunas fases de la producción. En estos programas también se realiza la retopología y el despliegue de *Uv's*. La utilización de los dos programas para estas tareas es indiferente, ya que ambos tienen prácticamente las mismas herramientas.

2) *Zbrush*: este software se especializa en escultura digital. Además, es bueno para realizar algunas retopologías automáticas para objetos estáticos.

-Texturizado:

3) *Substance Painter*: programa que se utiliza para el texturizado de los *assets*, sin embargo, también se puede utilizar para hacer un *bake* de texturas.

4) *Photoshop*: principalmente se utiliza para la edición de fotos, sin embargo, Photoshop es un software imprescindible para el retoque de las texturas y creación de la vegetación.

-Bake:

5) *Marmoset*: principalmente es un visualizador 3D en tiempo real, pero su uso más común es para hacer los mapas de *bake*. Esto es debido a que podemos ver cómo queda el *bake* en tiempo real.

-Motor de videojuegos:

6) *Unity*: es un motor de videojuegos en el que se ha hecho todo el montaje del escenario y *set dressing*[10].

-Edición de video:

7) *Adobe Premiere*: editor de videos en el que se ha montado la cinemática final.

-Referencias:

8) *PureRef*: permite generar bibliotecas de imágenes pudiendo agruparlas y ordenarlas al gusto del usuario. Es muy utilizado por los artistas digitales para visualizar las referencias del proyecto.

[10] Set dressing: Proceso de montaje del escenario.

6. DESARROLLO DE PROYECTO

6.1 PREPRODUCCIÓN

Para poder desarrollar un proyecto de esta envergadura, hay que seguir un proceso de producción de forma planificada y con las técnicas adecuadas, algunas de ellas ya descritas anteriormente.

6.1.1 *CONCEPT ART*.

Se refiere a un tipo de ilustración utilizada desde películas de animación a videojuegos e intenta transmitir una idea, estética o forma de cualquier elemento antes de su acabado como producto final. Es decir, anticipa el aspecto del producto final plasmando visualmente la idea clave o el concepto en la en la etapa de preproducción. (Arteneo, 2020).

Sin embargo, una obra de *concept art* no necesariamente tiene que ser una buena en sus dos aspectos, el concepto y el arte, como a diferencia de la ilustración. El concepto es la idea mediante la cual se define y contextualiza el elemento en una realidad concreta. El arte representa este concepto con una estética que se utilizará de referencia en la fase de producción. Así pues, un buen *concept art* es aquel que aporte una mayor y precisa información para la fase de producción.

De este modo, los trabajos de concept art se clasifican en diferentes subgrupos según su temática (Jaime, 2015):

Character design: está enfocado en el desarrollo de cualquier tipo de personaje (humano o no), pudiendo variar el enfoque entre una mayor personalidad o una estética más impactante (ver Fig. 15).

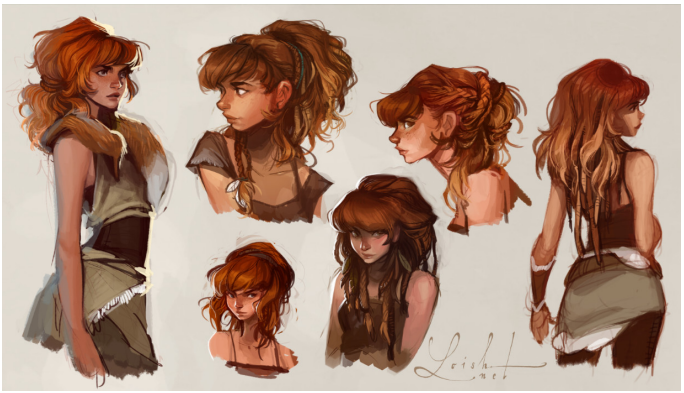

Fig. 15. *Character design* de Lois van Baarle.
Recuperado de: https://cutt.ly/NWv4vSi

Environment design: engloba todo tipo de escenarios y ambientaciones centrándose en aspectos como la iluminación, la climatología, la época o la cultura.

Key Art: se trata de determinados tipos de concept que narran una parte específica de una historia y acogen valores más argumentales aportando aspectos como la comprensión cinematográfica, la fotografía o la composición.

Prop design: representa un aspecto concreto de una ambientación mayor. Normalmente aportan de forma explicativa el funcionamiento de algún tipo de mecanismo (ver Fig.16).

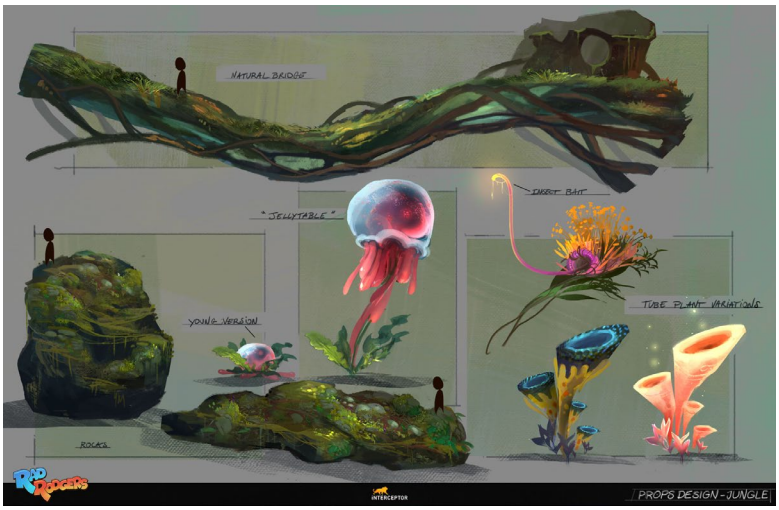

Fig. 16. Prop design de Florian Coudray.
Recuperado de: https://cutt.ly/XWv4DJV

En mi caso, partiendo de las indicaciones del concurso que permiten utilizar el arte conceptual de otro artista, me he servido de dos *concepts* de *Environment design* del artista Svetlin Velinov para definir la ambientación de mi escenario.

Ambos *concepts* aluden a un bosque pantanoso de un mundo fantástico y comparten la misma ambientación y estética similar (ver Fig.17). Esto permitirá integrar las piezas principales que componen las obras de una forma homogénea.

Fig. 17. *Concept arts* seleccionados para el desarrollo del proyecto. Autor: Svetlin Velinov. Collage propio.

6.1.2 BIBLIOTECA DE REFERENCIAS

Una de las piezas fundamentales para comenzar el proceso de desarrollo es recopilar imágenes de referencia que servirán como ayuda durante todo el trabajo. Aunque dispongamos de un escenario ya diseñado, es necesario disponer de referentes de los que partir, haciendo partícipe este proceso dentro del proyecto.

Esta fase se ha llevado a cabo en el software de PureRef, el cual permite agrupar múltiples imágenes y ordenarlas como disponga el usuario. Se trata de una ventana flotante en primer plano que ofrece una biblioteca de fotos rápida y ágil para artistas digitales.

Así pues, agrupé las imágenes seleccionadas teniendo en cuenta el tipo de objeto o factor diferencial. Un grupo fue enfocado al esqueleto, tanto cráneos como

costillas y vértebras, otro en árboles centrándose en ramas y raíces, otro de rocas y gemas y, por último, diferentes tipos de vegetación (ver Fig. 18).

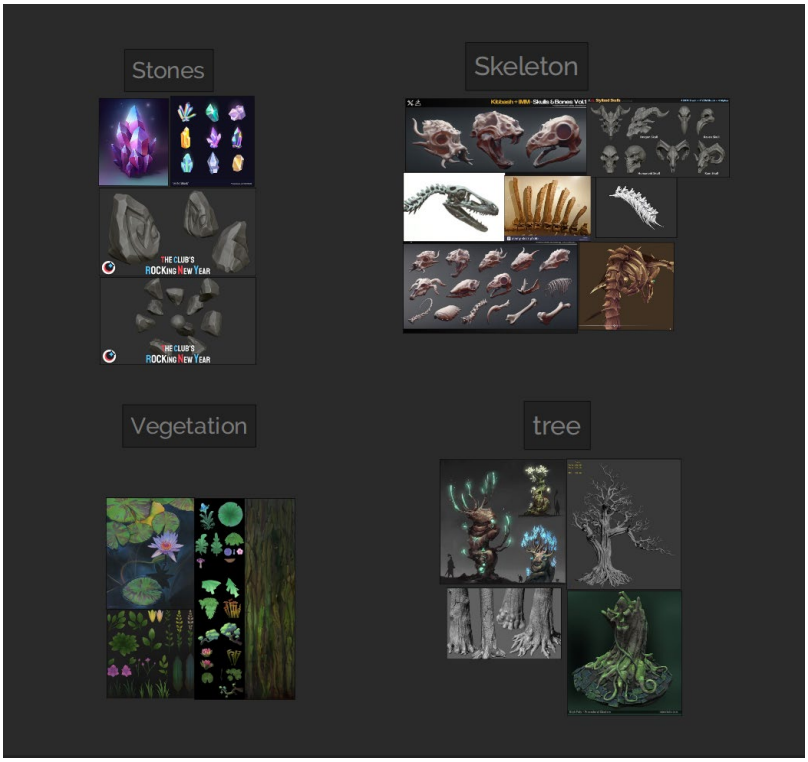

Fig. 18. Imágenes de referencia en Pureref. Elaboración propia.

6.1.3 *BLOCKING* Y CONCEPTUALIZACIÓN

El proceso de modelado comienza con el desarrollo de un boceto 3D, denominado *Blocking*, compuesto por estructuras básicas que permite ubicar los elementos y personajes en una escena determinada. Es una técnica que permite variar los tamaños y proporciones de una forma rápida y eficiente, sin precisar de un alto procesamiento del dispositivo utilizado.

Este borrador debe ofrecer a los artistas 3D una idea clara y definida de la dirección y enfoque del proyecto para que los siguientes procesos se lleven a cabo de una manera eficaz y homogénea.

Así pues, en lo que se refiere al proyecto, se ha realizado un estudio de los *concepts art* para poder definir una vista alzada de la distribución del escenario antes de comenzar con el *blocking* 3D (ver Fig.19). En este plano también se han incluido diferentes cámaras para poder tener una idea de los renders y la cinemática final. De este modo, se han ubicado los objetos partiendo del recorrido de la cámara que desvelará de forma sucesiva las piezas principales del escenario. Por lo tanto, el recorrido inicia en la frondosidad del bosque hasta alcanzar el árbol con portales ubicado en la orilla de una laguna donde se encuentra el esqueleto del dinosaurio.

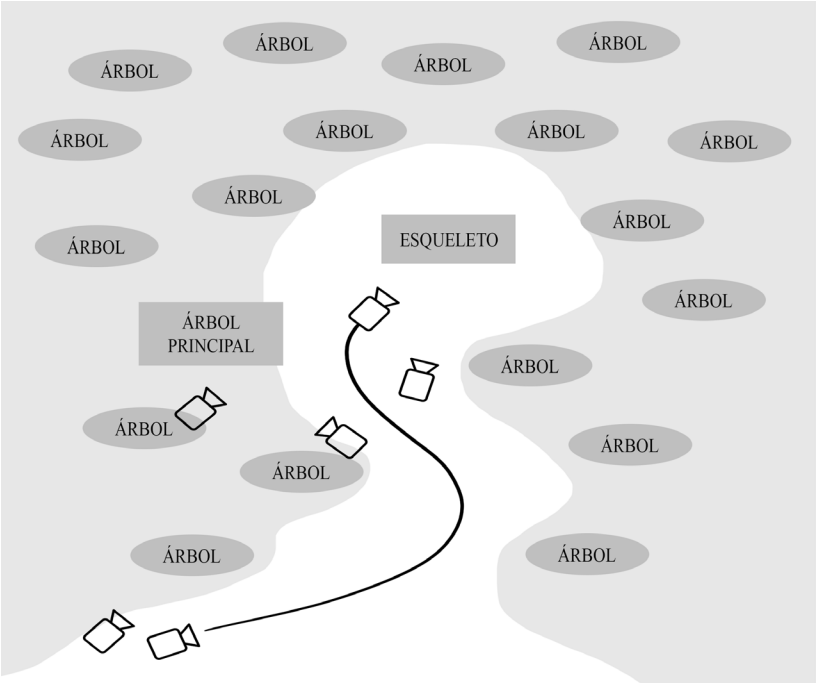

Fig. 19. Vista cenital de la distribución del escenario. Elaboración propia.

A continuación, se ha desarrollado el *blocking*, partiendo de un terreno de 50 x 50 m, con los diferentes modelos para construir la estructura general compuesta por los árboles, portales, el esqueleto, gemas, rocas, nenúfares y alguna planta, los salientes de tierra y lodo y el agua.

La estructura del *blocking* suele componerse de primitivas como cubos, esferas, cilindros, etc. Sin embargo, como se puede apreciar en la imagen, para algunos de los objetos, como el esqueleto o los árboles, se ha realizado una pequeña escultura digital mediante el modo *Sculpt* de Blender para poder definir de forma general los volúmenes de estas piezas y poder partir de éstas en el proceso de modelado (ver Fig. 20).

Fig. 20. Blocking del proyecto. Elaboración propia.

6.2 PRODUCCIÓN

6.2.1 MODELADO

El escenario consta de diferentes piezas independientes en el modelado: partes orgánicas (esqueleto, árboles, rocas, gemas, escaleras) que se han llevado a cabo mediante escultura digital en Zbrush y estructuras (portales) realizadas a través del modelado tradicional en Blender.

6.2.1.1 ESCULTURA DIGITAL

Esqueleto.

El proceso de escultura del esqueleto ha partido de la malla base realizada en el *blocking*, dónde se definieron los volúmenes y tamaños de todos los objetos. Una vez importado el fbx[11] del objeto desde Blender a Zbrush, se determinaron con más exactitud las formas y acabados que componen las piezas.

A diferencia del cráneo, que ya poseía los volúmenes principales, las vértebras del esqueleto se volvieron a modelar en Zbrush mediante cilindros y haciendo uso de booleanas para poder crear el agujero central que las une mediante la espina dorsal (ver Fig.21).

En cuanto a las costillas y dientes, partieron de planos y uso de modificadores para poder darle la forma deseada. Por otro lado, únicamente se definió el lado izquierdo de éstos ya que, una vez realizada la optimización de polígonos (retopología) y despliegue de *Uv's*, se utilizará el modificador "*Mirror*" para duplicar de manera espejo los objetos al lado derecho y así agilizar el trabajo.

Posteriormente, se realizaron los acabados finales de manera homogénea a todas las piezas haciendo uso de diferentes pinceles y *alphas*[12] de Zbrush que permiten dar un aspecto de hueso.

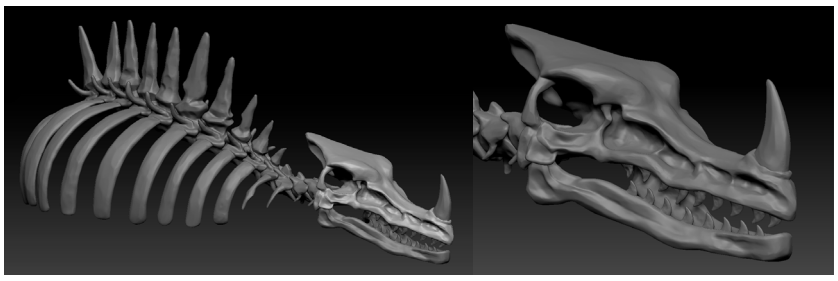

Fig. 21. Escultura del esqueleto en Zbrush. Elaboración propia.

[11] Fbx: es un formato de archivo que se encarga de mantener los elementos, funcionalidad e información del objeto 3D original.
[12] Alpha: permiten darle detalles y micro detalles a un modelo.

Árboles

Para la creación de los árboles, se ha aplicado el mismo procedimiento. A partir de una malla simple se ha realizado el detalle en Zbrush (ver Fig.22). Además, se tomó la decisión de llevar a cabo únicamente cuatro troncos: el árbol principal con los portales y tres que se esculpirán con diferentes volumetrías para poder reutilizarlos por todo el escenario sin que se aprecie un patrón de repetición. Para ello, se duplicarán y rotarán las veces que sean necesarias.

Por otro lado, se llevaron a cabo las ramas y raíces que pertenecen al árbol principal. Éstas se desarrollaron de forma individual asegurando una buena conexión con el tronco para que simulen un mismo objeto y, del mismo modo que los troncos, se utilizarán las veces necesarias para romper con la repetición y optimizar el trabajo.

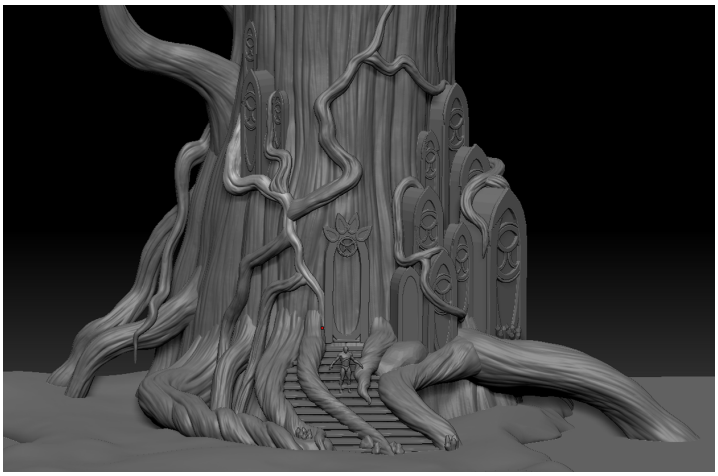

Fig. 22. Escultura del árbol principal en Zbrush. Elaboración propia.

En cuanto a los acabados de los árboles, se hizo uso de *alphas* en Zbrush para generar las vetas de los troncos, ramas y raíces.

Rocas y gemas

El modelado de rocas y gemas ha seguido el procedimiento general: un modelado base con primitivas en Blender y posterior preforma en Zbrush para ajustar el detalle (ver Fig. 23). Por otro lado, también se optimizará su uso, de modo que se han generado un total de cuatro gemas y seis piedras que se distribuirán por el escenario.

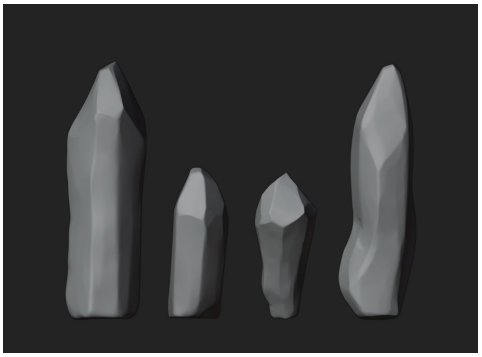

Fig. 23. Escultura de las gemas en Zbrush. Elaboración propia.

Escaleras.

Para el esculpido de las escaleras se ha partido de la malla base compuesta por cubos. Además, al encontrarse ubicadas en el árbol principal y tener diferentes raíces por encima se decidió generar diferentes roturas y grietas para simular la erosión del crecimiento de las raíces sobre estas. Por lo que para generar estos detalles fue necesario importar tanto las escaleras como las raíces para controlar la ubicación exacta donde se debían encontrar dichas roturas (ver Fig. 24).

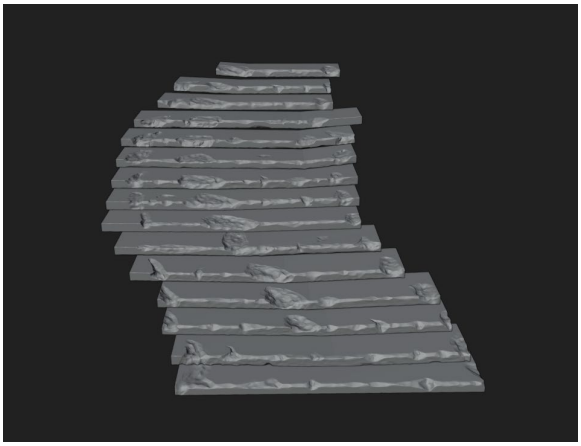

Fig. 24. Escultura de las escaleras en Zbrush. Elaboración propia.

6.2.1.2 MODELADO DE ESTRUCTURAS

Portales.

Para la creación de los portales se ha utilizado el modelado tradicional como se ha explicado anteriormente (ver Fig.25). Como el resto de los objetos, también se ha optimizado su uso, de modo que únicamente se ha generado uno y duplicado y escalado diferentes veces. Así pues, los portales están compuestos por la estructura principal, el ornamento central y un plano al que, posteriormente, se aplicará emisión de luz y que también facilitará la diferenciación entre el portal activo y el resto.

Fig. 25. Modelado tradicional de estructuras. Elaboración propia.

6.2.2 RETOPOLOGÍA

Una vez tenemos los objetos modelados, se procede a realizar una retopología manual en 3Ds Max. Sin embargo, al tratarse de piezas esculpidas con un alto nivel de detalle, es decir, que se componen de un número elevado de polígonos, deberían ser optimizados antes de comenzar con el proceso. Esto se debe a que los programas de modelado tradicional no son capaces de soportar la misma cantidad de geometría. Para ello, se han preparado las piezas para la retopología mediante herramientas como *Decimate* o *ZRemesh* de Zbrush.

Así pues, la herramienta *Decimate* permite reducir el número de polígonos de una escultura de alta resolución sin perder los detalles. Este plugin de Zbrush da como resultado una malla compuesta de tris (triángulos) con notable reducción de polígonos. Sin embargo, a diferencia de la herramienta *Zremesh*, esta reducción puede ser más alta puesto que, al triangulizar, consigue mantener mucho mejor los detalles de la escultura, con menos geometría. De este modo, la pieza resultante de este proceso será el modelo que utilizaremos en el programa de modelado para llevar a cabo la retopología manual.

Como resultado de este proceso se han obtenido las siguientes cifras de polígonos: El esqueleto con 2,3 millones, el árbol principal con 5,5 millones, el resto de los árboles entre 222 y 258 mil, las rocas entre 108 y 34 mil, las gemas entre 11 y 25 mil y las escaleras con 3.2 millones.

Por otro lado, el plugin *Zremesh* genera una reducción de polígonos mediante *quads* (cuadrados), por lo que es más complicado mantener con poca geometría toda la forma y los detalles. Esto genera una organización en la geometría, pero no lo suficientemente limpia como para pasar a la siguiente fase. Por esta razón se procede a realizar la retopología manual de los modelos partiendo de esta geometría generada.

Sin embargo, este proceso únicamente se ha aplicado al esqueleto ya que, para el resto de los objetos es más simple partir de primitivas para realizar la retopología que arreglar la generada por la herramienta *Zremesh*. De este modo,

tras realizar este proceso se ha obtenido un total de 95.300 polígonos, una diferencia de 2.2 millones de polígonos de optimización.

Por último, se ha realizado la retopología manual de los objetos para optimizar todo lo posible, pero sin perder la forma original de los objetos (ver Fig.26). En el caso del esqueleto se ha obtenido un número final de 39 mil polígonos, en el árbol principal 12 mil, el resto de árboles entre 600, la escalera con 10 mil, las rocas entre 300 y 600 y las gemas alrededor de 100 polígonos.

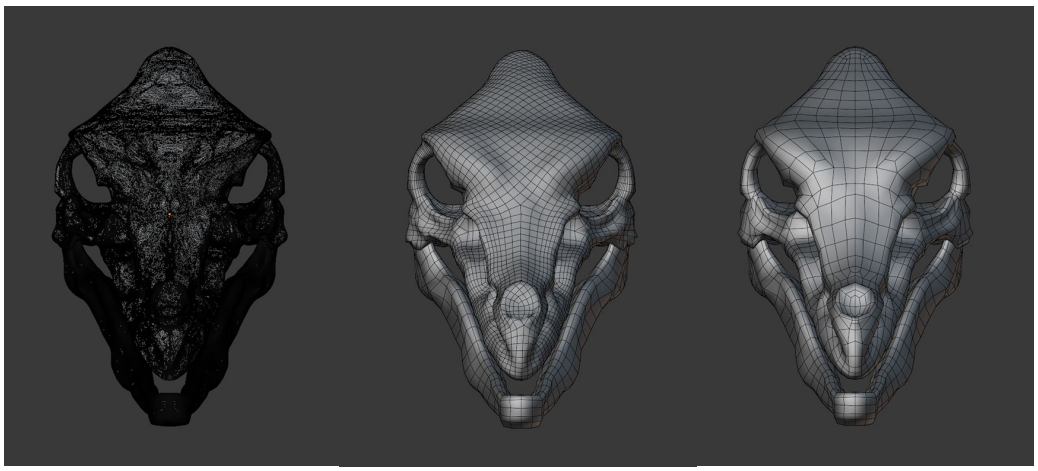

Fig. 26. Modelo optimizado mediante *Decimate, Zremesh* y retopología manual. Elaboración propia.

6.2.3 MAPEADO UV, BAKE Y TEXTURIZADO

Mapeado de Uv's

El mapeado es la siguiente fase a la retopología, donde se ordena de forma correcta y óptima el espacio de las coordenadas *Uv*.

En este proyecto se ha decidido realizarlo en mapas 4K, es decir 4096 x 4096 pixels teniendo en consideración que el escenario engloba un concurso y Trabajo Final de Grado además de que terminará siendo una pieza de porfolio, con lo que la calidad debe ser la máxima posible.

Las diferentes piezas del escenario se han diferenciado en *Texture Sets*, es decir en diferentes sets de texturas de cada material, por lo que cada material tendrá un espacio de *Uv's*. Así pues, en lugar de tener todos los objetos del escenario en un mismo set de *Uv's*, se han dividido en: esqueleto, árboles, ramas y raíces, piedras, gemas y escaleras.

Sin embargo, el esqueleto se ha separado en tres grupos diferentes: vértebras y dientes, costillas y cráneo. Puesto que, al tratarse de la pieza principal y su tamaño, no se puede permitir una baja calidad de texturizado lo cual haría que la textura quedará pixelada. Por otro lado, una vez sacadas las *Uv's* de los colmillos, se ha aplicado el modificador "*Mirror*" en el objeto, como se ha comentado anteriormente. Con ello conseguimos un mayor espacio en nuestra textura, ya que las *Uv's* de los colmillos duplicados se encontrarán solapadas, unas encima de otras compartiendo la misma textura. Esto no es un problema ya que es raro que se puedan ver las dos partes a la vez del esqueleto y ayuda mucho en cuanto a optimización se refiere (ver Fig. 27).

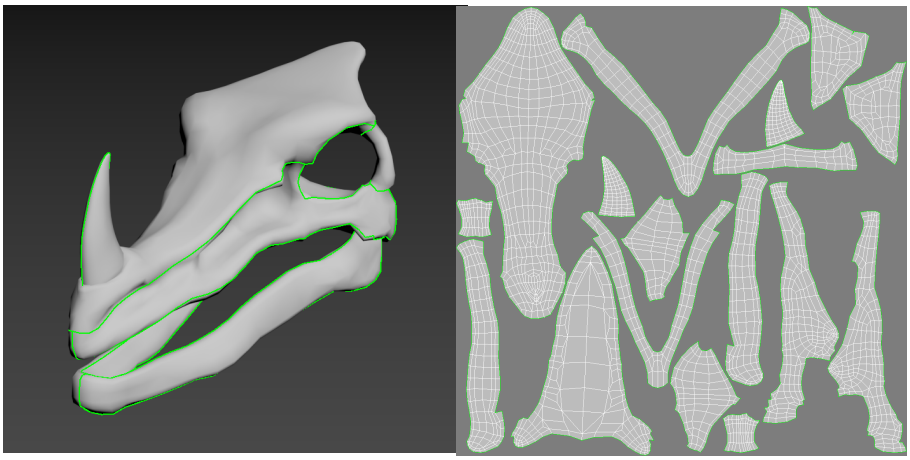

Fig. 27. *Seams* de despliegue de *Uv's* del cráneo y *Texture Set* del cráneo. Elaboración propia.

En cuanto a los árboles, se ha decidido proporcionar el espacio de *Uv's* de forma independiente ya que para éstos se hará uso de una textura tileable que se aplicará tanto a los troncos como a ramas y raíces. Que dispongan de un set independiente garantiza que la textura funcione de forma correcta. De lo contrario, podría generar imperfecciones ubicadas en los cortes de las *Uv's*.

Así pues, se han realizado los *seams* o cortes en la geometría en lugares que sean lo menos visibles posible con el fin de evitar efectos no deseados en las

texturas. A continuación, tras realizar los cortes adecuados y despliegue de *Uv's*, se han ordenado de forma adecuada dejando un espacio entre las diferentes islas de *Uv's* con el fin de aprovechar el mayor espacio posible.

Por último, se ha añadido un segundo canal de *Uv's* para generar los *Lightmaps* en el motor a tiempo real. Estos mapas almacenan la información de luz y sombra dentro de la escena que fue calculada en el proceso de *lightmass* o *baked lightmap*. Este proceso se explicará con más rigurosidad en el apartado de iluminación.

Baking

Como se mencionó anteriormente, el *baking* hace referencia al proceso de creación de los diferentes mapas que se proyectan sobre los modelos *low poly* (ver Fig. 28). Este proceso se ha llevado a cabo en Marmoset ya que genera resultados con calidad de forma instantánea.

De este modo, se han generado los mapas de Normales, el *Curvature Map*, *Ambiet Occlusion, Position* y *Thickness Map*.

Para llevar a cabo este proceso se han importado tanto la malla *low poly* como la de *high poly* con su respectiva nomenclatura en el espacio de trabajo de Marmoset y se ubican en la sección destinada del *baker* del menú. En este apartado se especifican las propiedades y mapas para el proceso de *bakeado*.

Este proceso se ha realizado en todas las piezas que componen el escenario exceptuando la vegetación, el suelo y el agua.

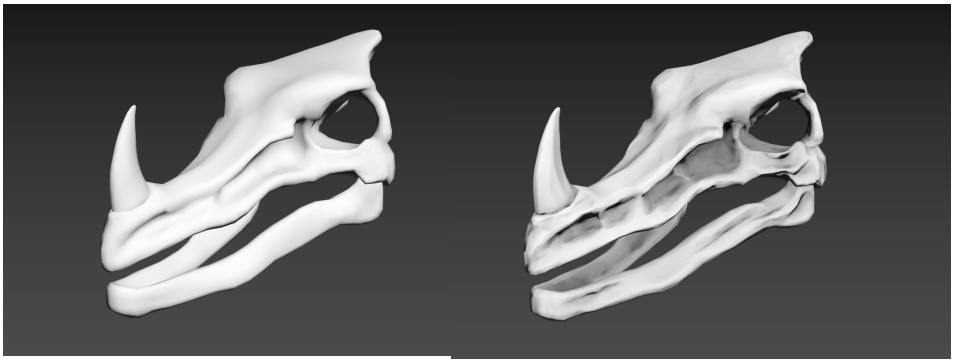

Fig. 28. Comparación del modelo sin y con los mapas de normales.
Elaboración propia.

Texturizado

Como se ha comentado anteriormente, el proceso de texturizado se ha llevado a cabo con Substance Painter. Este software permite texturizar directamente sobre la malla tridimensional permitiendo agilizar el flujo de trabajo y obteniendo de forma directa resultados visibles.

En primer lugar, se ha llevado a cabo el texturizado del esqueleto. El hueso se ha organizado en diferentes capas que en su conjunto conforman un resultado de hueso desgastado y mohoso. Para ello, se ha establecido una capa base que se aplica a todo el objeto de forma automática sin quedar geometría por rellenar. Seguidamente, se han creado diferentes capas dónde se han utilizado múltiples pinceles y colores. Además, se ha hecho uso de algunos de los mapas generados en el *bake* como el *Ambient Oclusion* o *Curvature* que, junto a máscaras, permite teñir las luces y sombras de una forma más ágil (ver Fig. 29).

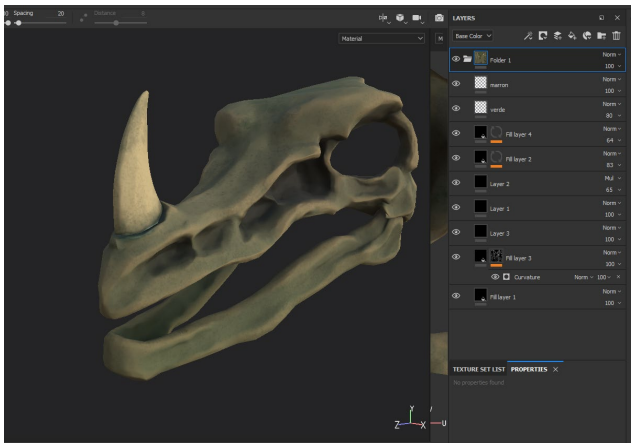

Fig. 29. Texturizado del cráneo en Substance Painter. Elaboración propia.

En cuanto a los árboles, se ha desarrollado una textura utilizando diferentes capas y materiales proporcionados por Substance Painter. Una vez generada esta textura, se ha modificado en Photoshop haciendo uso de las herramientas como *Tampón de Clonar* o *Parche* para poder transformarla en una textura tileable y aplicarla tanto en troncos, ramas y raíces (ver Fig.30).

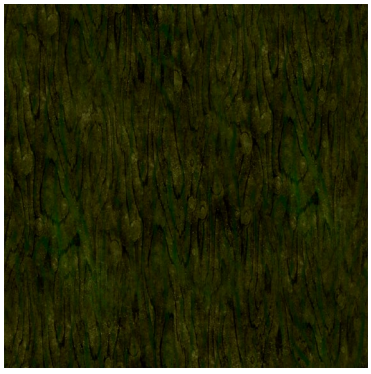

Fig. 30. Textura tileable de árbol. Elaboración propia.

Las piedras han seguido el mismo procedimiento que el esqueleto. Se ha utilizado un color base y diferentes capas para crear humedades y hongos en la superficie. Sin embargo, para las gemas únicamente se ha utilizado el color base y un degradado de abajo a arriba puesto que va a ser un material emisivo y pequeños detalles no llegarán a ser visibles. Además, se ha generado un nuevo mapa emisivo en Photoshop para generar diferentes niveles de emisión en éstas.

Por otro lado, se han generado las texturas que cubrirán el suelo utilizando la técnica de *Photobash* en Photoshop (ver Fig.31). Esto consiste en combinar diferentes fotografías, como si se tratara de un collage, para conseguir resultados realistas. Así pues, se han creado mediante herramientas de Photoshop como los modos de fusión, dos texturas de césped y lodo que se combinarán en la superficie del suelo.

Fig. 31. Textura tileable de césped generada mediante *Photobash*.Elaboración propia.

Por último, se ha creado un *God Rays* o rayo de luz mediante degradados en Photoshop para simular la luz que entra entre las copas de los árboles. Esto se ha llevado a cabo debido a que este efecto no se puede generar mediante la iluminación de Unity y se ha tomado la decisión de falsearlos.

6.2.4 VEGETACIÓN

La creación de vegetación para videojuegos es un proceso delicado ya que se debe tener muy en cuenta la cantidad de geometría que poseen cada una de las piezas. Esto se debe a que cada una de las plantas se va a duplicar numerosas veces y, en el caso de que poseyera una alta carga poligonal, afectaría al rendimiento del escenario.

Dicho esto, esta fase se ha comenzado con el texturizado en Photoshop de los diferentes tipos de hoja que, posteriormente, se montaron en Blender para crear el conjunto de la planta (ver Fig.32). Así pues, se han creado dos atlas con una

resolución de 2048 x 2048 pixels que incluyen tres tipos de nenúfares, diferentes tallos de césped, dos tipos de hojas, un trébol y musgo.

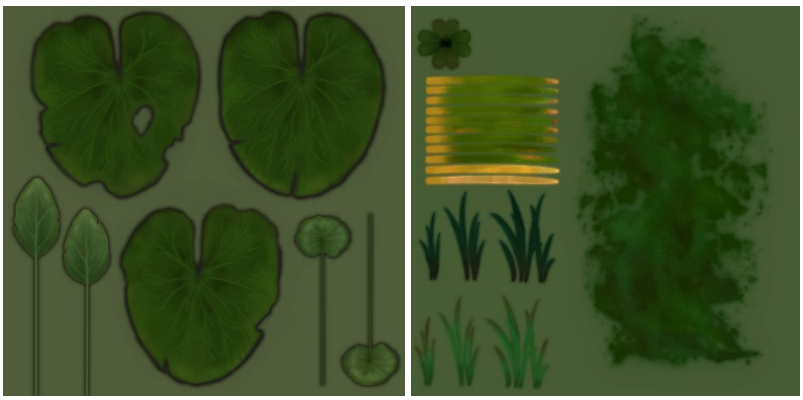

Fig. 32. Atlas de vegetación. Elaboración propia.

Una vez texturizado, se ha duplicado las diferentes piezas por debajo de la principal y se ha añadido un desenfoque gaussiano. Esto provoca que al aplicar el mapa de transparencia o *alpha* el motor se traduzca en un degradado y no en un corte recto que generaría dientes de sierra o *aliasing*.

A continuación, se han generado los mapas de normales y transparencia mediante herramientas de Photoshop (ver Fig.33). Los mapas de transparencia se han producido creando una máscara de recorte blanca sobre estas y un fondo negro. Por otro lado, los mapas de normales mediante un plugin capaz de generarlos a partir de una imagen 2D.

Fig. 33. Mapa de transparencia y *Normal Map* del Atlas 1. Elaboración propia.

Llegados a este punto, se han aplicado estas texturas en planos creados en Blender con el fin de separar cada una de las piezas y montar el conjunto de la planta con cada elemento (ver Fig.34).

Posteriormente, se ha llevado a cabo un proceso necesario para la correcta visualización de las plantas en el motor que corrige la orientación de las normales unificándolas. Para ello, se ha hecho uso de una esfera que envuelve al objeto y proyecta sus normales al conjunto de planos que componen la planta.

Por otro lado, los juncos han sido modelados mediante el modelado tradicional puesto que consideré que el volumen de esta no podría no podía conseguirse mediante planos como el resto de las plantas.

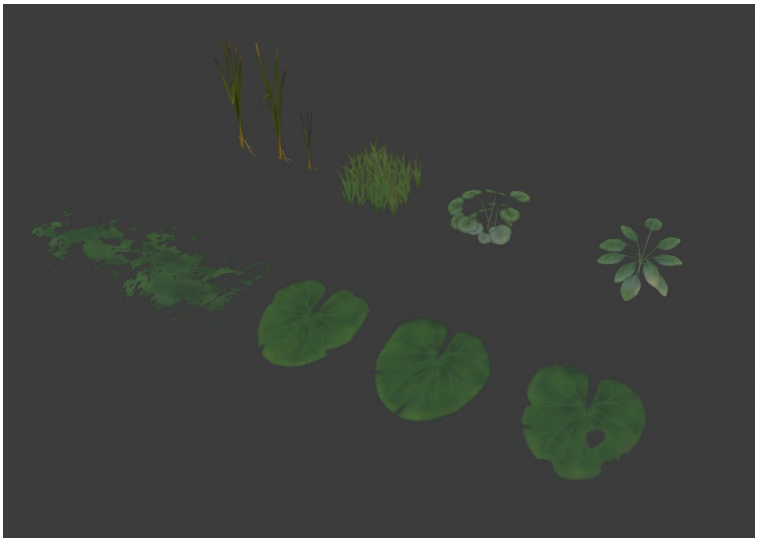

Fig. 34. Montaje de la vegetación en Blender. Elaboración propia.

6.2.5 MONTAJE

Una vez generados todos los *assets* es el momento de ubicarlos en la escena.

Fig. 35. Montaje de piezas principales del escenario y creación de *Terrain*. Elaboración propia.

Para ello, se ha seguido el esquema que se desarrolló en la fase de *blocking* y se han posicionado las piezas principales, es decir, el esqueleto y el árbol. Por otro lado, se han colocado el resto de árboles y reutilizado tanto los troncos como las raíces y ramas para cubrir todo el escenario (ver Fig.35).

Seguidamente, se han creado las diferentes cámaras que mostrarán el resultado final para ajustar la composición de los planos antes de continuar con la creación del terreno, vegetación, agua y partículas.

Una vez compuestos los planos de forma correcta, se ha utilizado la herramienta *Terrain*, proporcionada por el software de Unity, para producir el terreno de la escena. Este plugin genera un plano, en mi caso de 50 x 50 m, y permite ajustar la altura o el aspecto e incluir diferentes tipos de vegetación. De este modo, se ha modificado el *Terrain*, a través de diferentes pinceles facilitados por el editor, para crear los salientes de lodo y tierra que rodean los árboles.

Además, esta herramienta permite combinar diferentes texturas a la superficie incluyendo sus mapas de normas. Así pues, se han importado las texturas tileables realizadas anteriormente junto con su *Normal Map* y se ha texturizado utilizando el césped como textura base y el lodo en la orilla del agua.

A continuación, se incluyeron los diferentes tipos de vegetación. Para ello, se importaron los modelos en Unity y, a pesar de que se podrían haber incluido en el editor del *Terrain* para agregarlas mediante un pincel, se decidió añadirlas de manera independiente. Esto se debe a que, mediante el editor, no se consiguió el resultado esperado ya que la distribución de las plantas era aleatoria y poco precisa. Además, tampoco suponía ninguna diferencia en cuanto a optimización.

Por otro lado, en cuanto a los nenúfares, se posicionaron en la escena sobre un plano base que posteriormente se sustituyó por el *shader* de agua (ver Fig.36).

Fig. 36. Montaje con todos los modelos del escenario.
Elaboración propia.

Sin embargo, cabe resaltar que la fase de montaje es un proceso iterativo, es decir, que a pesar de que se realiza en las últimas fases del proyecto es importante ir implementando los *asstets* realizados para conseguir una mayor integración en el escenario. Esto permite percatarse de fallos que pueden afectar al estilo y corregirlos antes de que acabe la fase de desarrollo.

6.2.6 *SHADER GRAPH*

El *Shader Graph* es una herramienta proporcionada por Unity que permite trabajar con un canal de renderizado codificable que funciona mediante nodos y permite definir el comportamiento de estos mediante el editor. Con este plugin se pueden generar efectos artísticos o especiales como visión térmica, nieve, agua, etc. (Unity Learn, s. f.).

En este proyecto se ha hecho uso de un *shader* de agua libre de derechos adquirido en la *Asset Store*[13] y un *shader* de portal mágico creado a partir de un video tutorial de YouTube donde explica paso por paso cómo llevarlo a cabo.

***Shader* agua.**

Para la realización del agua se ha importado el *shader* adquirido dentro de la escena de Unity y se han variado diferentes parámetros para conseguir el estilo de agua deseado. Así pues, se han modificado los aspectos que afectan al color, la cantidad, la altura y la frecuencia de las olas, el reflejo, la transparencia y la velocidad de movimiento. Todos los parámetros se han corregido con el fin de conseguir un agua estancada, aunque con cierto nivel de movimiento para simular la vida subacuática. Además, se han incluido dos colores de tonalidad verde dependiendo de la profundidad del agua y, gracias a la transparencia configurada, en algunas zonas del escenario se puede apreciar el fondo del pantano (ver Fig.37).

[13] Asset Store: biblioteca de **assets** comerciales y gratuitos creados por Unity Technologies y miembros de la comunidad.

Fig. 37. Resultado del *Shader* de agua. Elaboración propia.

Shader **portal.**

La creación de *shader* del portal se ha llevado a cabo siguiendo el tutorial del canal de YouTube Gabriel Aguiar Prod: "*Unity VFX Graph - Magic Orb Effect Tutorial"*[14]

Se han seguido todos los pasos realizados hasta llegar a la creación de este (ver Fig.39). Se han creado diferentes nodos para generar partículas que afectan a su color, cantidad, tiempo de vida, velocidad y emisión. Además, se ha delimitado de movimiento dentro de una esfera transparente para que las partículas únicamente se muevan en este rango y se ha escalado para que encaje dentro del portal.

Por otro lado, se ha generado otro sistema de partículas mediante nodos para generar la superficie base del *shader* y se le ha dotado de color y transparencias.

De este modo, se han creado dentro de *Shader Graph* tres estructuras principales (ver Fig.38). La primera donde se delimitan los aspectos del sistema de partículas que genera los rayo en movimiento, la segunda donde se define la forma y margen de movimiento, y, la última en la que se determina el color base del portal y su transparencia.

[14] https://www.youtube.com/watch?v=7bMOhNUA1bI&ab_channel=GabrielAguiarProd.

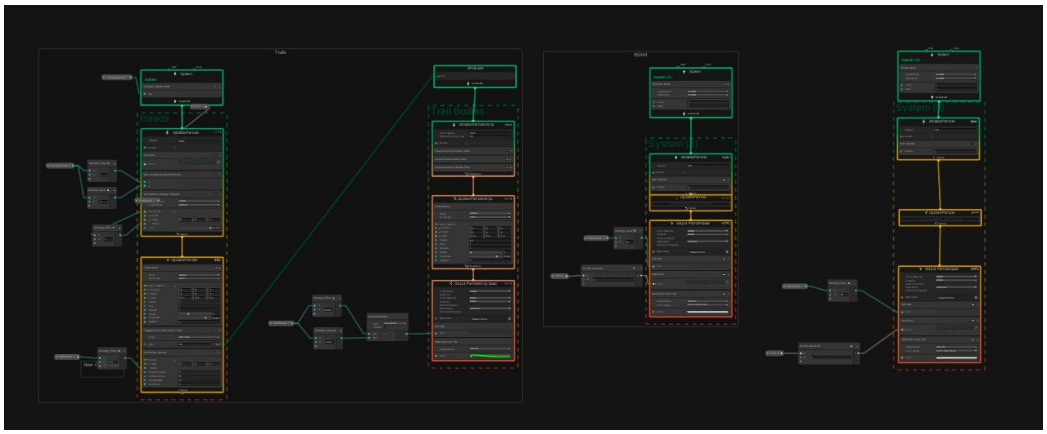

Fig. 38. Estructura del *Shader* del portal. Elaboración propia.

Fig. 39. Resultado *Shader* del portal. Elaboración propia.

6.2.7 PARTÍCULAS

Los sistemas de partículas de Unity son un conjunto de *sprites* o imágenes 2D que permiten generar efectos especiales como fuego, explosiones, disparos, humo, etc (Unity3D, s. f.). En este proyecto se han utilizado para añadir pequeñas motas de polvo flotando por toda la escena y para crear un ambiente mágico generando pequeñas chispas que brotan de las gemas.

Para ello, se han utilizado sistemas de partículas simples, ya que no suponen una gran complejidad. De este modo, para las motas de polvo se han modificado los parámetros que afectan al color, el tamaño, la emisión, el movimiento y la

distancia entre ellas. Además, se ha configurado la vida de cada una de las partículas con el fin de que aparezcan y desaparezcan en una ratio de entre cuatro y diez segundos (ver Fig.40).

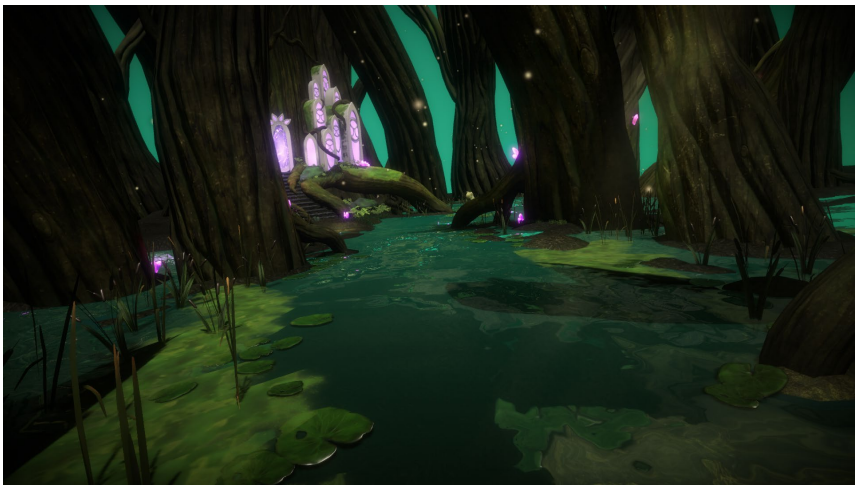

Fig. 40. Partículas de la escena. Elaboración propia.

Por otro lado, para las chispas de las gemas se han variado los mismos parámetros, aunque, a diferencia de las motas, el movimiento en este caso es vertical ascendente y la vida de las partículas oscila entre cuatro y cinco segundos (ver Fig.41).

Fig. 41. Partículas de las gemas. Elaboración propia.

6.3 POSTPRODUCCIÓN

El proceso de postproducción de este proyecto recoge los aspectos que permiten realizar los acabados finales del escenario. De este modo, a continuación, se explicará el flujo de trabajo que se ha seguido para iluminar, llevar a cabo el postprocesado de la imagen y renderizar los diferentes tiros de cámara necesarios para la participación del concurso.

6.3.1 ILUMINACIÓN

El software de Unity proporciona diferentes herramientas y tipos de luces para poder iluminar una escena. En primer lugar, cabe destacar que el motor de render diferencia entre cuatro tipos de luces (Unity3d, s. f.):

Directional Light que simula una luz de sol afectando a todos los objetos y permitiendo una iluminación base en la escena. Sin embargo, para este tipo de iluminación la posición en la escena no importa, puesto que se trata de luces distantes que existen de forma infinita, es decir, que afecta del mismo modo independientemente del lugar en el que se encuentre. De este modo, para poder variarla se hace uso de su rotación con el que los rayos de luz inciden en los objetos dependiendo de su orientación.

Point Light, ubicada en un punto del espacio y emite luz hacia todas las direcciones por igual.

Spot Light que tiene una posición específica y un rango sobre el cual incide la luz. Además, está limitada a un ángulo que genera de forma cónica la iluminación.

Area Light que se define en un rectángulo que emite luz de manera uniforme a los objetos generando sombras suaves sobre estos.

De este modo, para el proyecto se ha prescindido de la luz direccional puesto que generaba unas sombras duras que no acompañaban a la ambientación general. En su lugar, se han utilizado tanto Point como Spot Light que se han posicionado en diferentes lugares de la escena para poder guiar al jugador por el escenario hasta las piezas importantes (ver Fig.42).

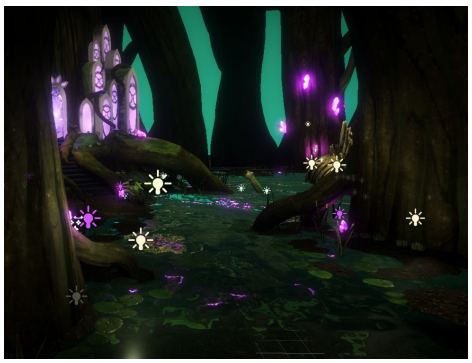

Fig. 42. Iluminación de la escena en Unity. Elaboración propia.

Por otro lado, se han hecho uso de luces Spot en las gemas para reforzar el material emisivo e incidir en las superficies cercanas.

Sin embargo, la iluminación en un motor a tiempo real ofrece diferentes modos distinguiendo entre *realtime, baked* y *mixed.* Estos modos se utilizan dependiendo del tipo de objeto sobre el que van a incidir, siendo objetos que tengan movimiento como los personajes, o estáticos, como lo puede ser un árbol o cualquier estructura. Así pues, el modo *realtime* calcula y actualiza la iluminación en cada *frame* durante el periodo de ejecución, es decir, va renovando la información de luz que incide sobre los objetos en movimiento. Por el contrario, el modo bake realiza un cálculo de iluminación previo al tiempo de ejecución, denominado *Bake*, y adhiere esa información a los *Light Maps* o mapas de luces que genera sobre los objetos estáticos sombras y luces de alta calidad sin sobrecargar el tiempo de ejecución. Por último, el modo *Mixed* combina los anteriores y permiten calcular la información de luz durante el tiempo de ejecución o precalcularlas previamente sobre los objetos estáticos (Unity3d, s. f.).

Por otro lado, cabe destacar que el modo *realtime* limita su uso a ocho luces, puesto que si se generarán más sería demasiada información para actualizar en cada *frame* y su rendimiento bajaría de forma notable. Por esta razón, se ha tomado la decisión de utilizar únicamente luces en modo *bake* puesto que todos los objetos que conforman el escenario son estáticos y no se requiere de una iluminación *realtime*. Por otro lado, esta decisión se refuerza debido a que los motores de videojuegos, excepto el actual Unreal 5, no son capaces de calcular el *Globa lIlumination o* iluminación global en tiempo real. Esto se basa en el cálculo de los rebotes de luz sobre las superficies de los objetos generando así una iluminación más realista.

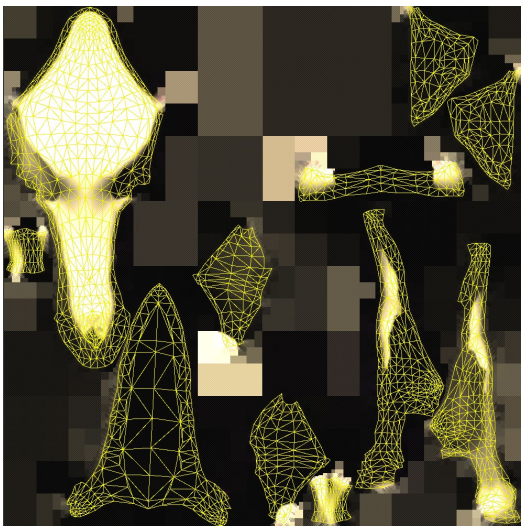

Fig. 43. *Lightmaps* generado en el *bake* de luces en Unity. Elaboración propia.

Así pues, se ha llevado a cabo el proceso de *Bake*. Para ello, se han definido todos los objetos de la escena como estáticos y se ha cambiado a modo *bake* toda la iluminación que se ha generado en la escena. De este modo, el proceso plasmará en el segundo canal de UV, creado anteriormente, toda la información de luz que incide sobre estos objetos (ver Fig.43).

Sin embargo, una vez generados los *Light Maps* no permite retroceder para modificar la iluminación. Por esta razón, se ha ido creando poco a poco las luces realizando el proceso de *bake* en cuatro fases diferentes: iluminación general, iluminación piezas principales, iluminación de relleno e iluminación para reforzar las gemas emisivas.

Por último, se han añadido los *God Rays* generados anteriormente para representar los rayos de luz que entran entre las ramas de los árboles (ver Fig.44). Para ello, se ha aplicado la textura a un plano junto con su mapa de transparencia y se han ubicado tanto en la laguna donde se encuentra el esqueleto como entre los árboles al comienzo de la cinemática.

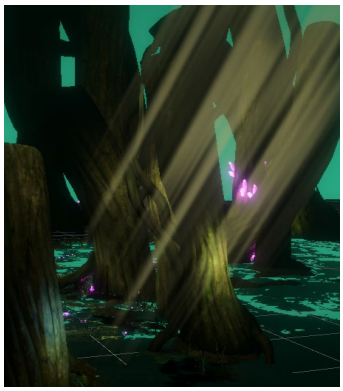

Fig. 44. *God Rays* en Unity. Elaboración propia.

6.3.2 POSTPROCESADO

Se trata de la fase final del proyecto donde se aplican diferentes filtros y efectos a la imagen de las cámaras que mostrarán el resultado final del videojuego. Estos efectos simulan las propiedades de las cámaras físicas y de cine permitiendo mejorar la calidad visual del proyecto.

Así pues, se ha procedido a la creación de un *Global Volume* o contenedor que recoge la información del postprocesado y se han creado los filtros que se han considerado necesarios.

Entre ellos se encuentra un *White Balance* o balance de blancos para equilibrar los niveles de color. *Shadows, Midtones and Highlights* para llevar a cabo un etalonaje con el fin de reforzar el ambiente generado a través de diferentes modificaciones en las tonalidades de brillos y sombras. Un Bloom que genera franjas de luz que se expanden desde los bordes de áreas emisivas generando un halo de luz en estos objetos. Y, por último, *Vignette* o viñeteado que, mediante un borde negro difuminado, permite reducir y centrar el campo de visión del espectador en el centro de la pantalla (UnityLearn, s. f.).

Para finalizar, se ha añadido desde el menú de *"Lighting-Environment*" una niebla con una tonalidad verde que permite apreciar la profundidad a la que se encuentran los objetos y ofrecer un ambiente húmedo a la escena (ver Fig.45).

Fig. 45. Comparación del escenario con postprocesado. Elaboración propia.

6.3.3 RENDERS Y CINEMÁTICA

Retomando los requisitos especificados en la base del concurso, la entrega debe incluir cinco renders y una cinemática del entorno final con una calidad mínima de 1920 x 1080 pixels.

Para ello, se ha hecho uso de las diferentes cámaras ubicadas durante el proceso y una *Dolly Camera* que, como en el cine, permite generar *travelling* o planos en movimiento siguiendo un rail (ver Fig.46).

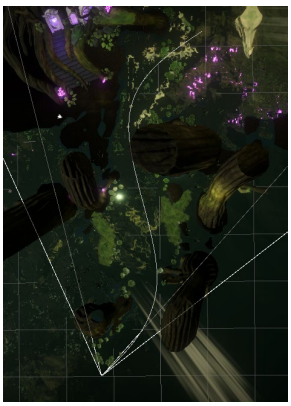

Fig. 46. *Dolly Cam* en Unity. Elaboración propia.

Por otro lado, para llevar a cabo el renderizado, se ha hecho uso de la herramienta *Recorder* de Unity que permite generar tanto renders de *frames* independientes como cinemáticas del resultado final.

6.3.4 MÚSICA

La música es capaz de generar diferentes sensaciones dependiendo tanto de su ritmo, instrumentos, tono, etc. Una cinemática sin música se convierte en algo monótono y sin ningún tipo de interés para el espectador. Por esta razón, se ha incluido la melodía "*Lord Of The Dawn"* de Jesse Gallagher de libre derechos adquirida en Youtube Library para acompañar al travelling.

Se trata de una música que evoca misterio, fantasía y aventura y acompaña a la cinemática descubriendo poco a poco las piezas principales. El montaje se ha llevado a cabo mediante el programa Adobe Premiere dónde únicamente se ha incluido el audio y se han creado fundidos al inicio y al final del video.

7. RESULTADOS FINALES

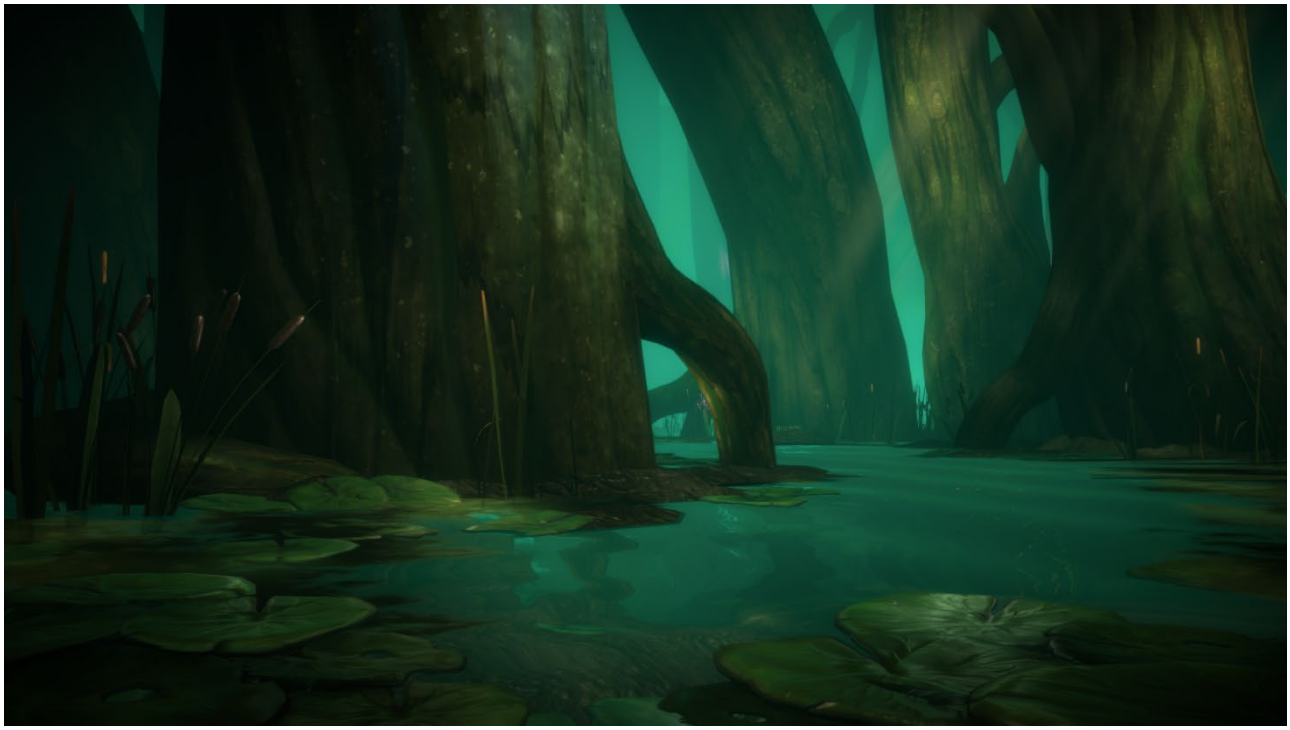

Fig. 47. Render final uno del proyecto. Elaboración propia.

Fig. 48. Render final dos del proyecto. Elaboración propia.

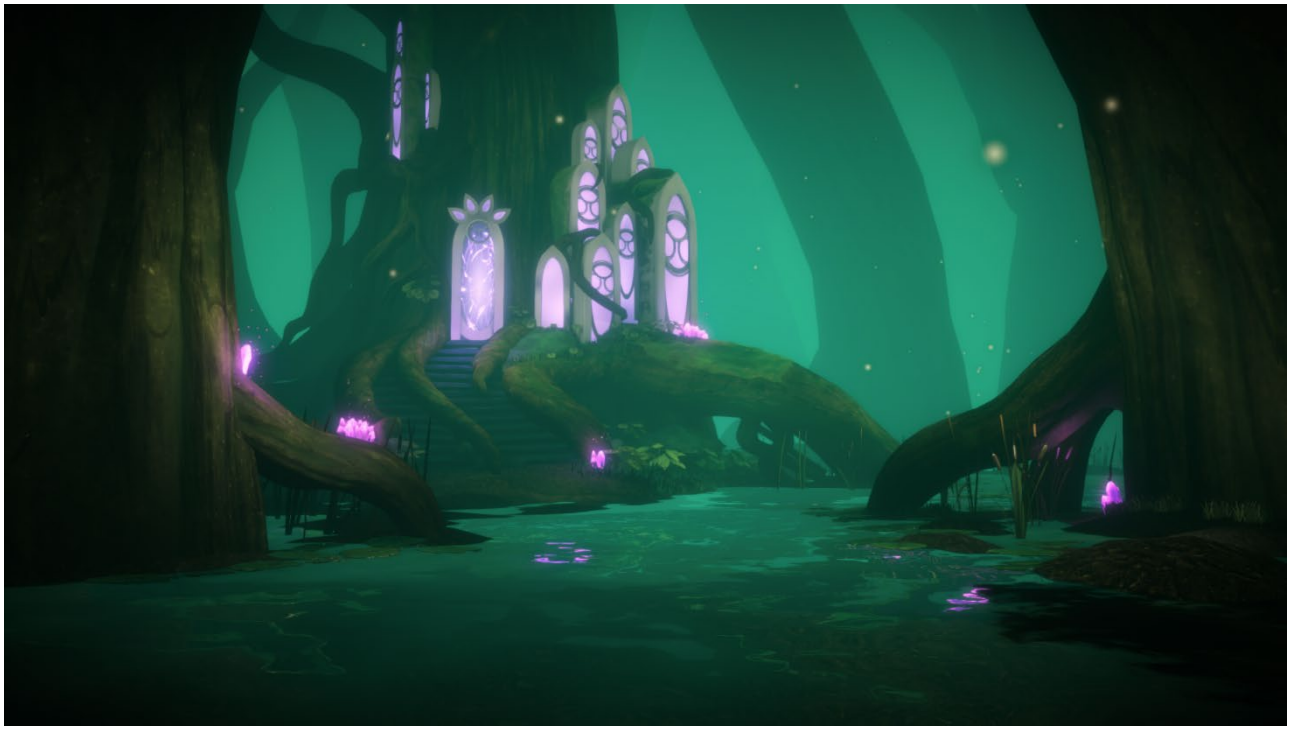

Fig. 49. Render final tres del proyecto. Elaboración propia.

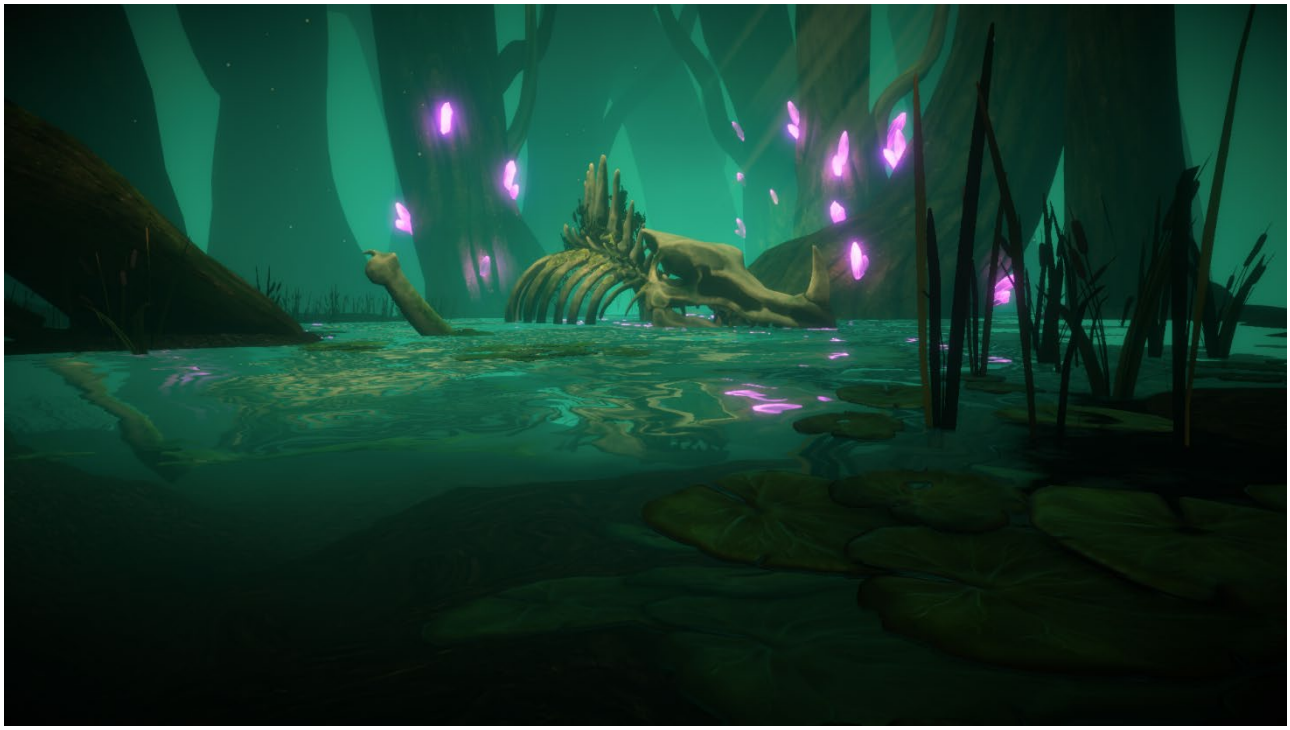

Fig. 50. Render final cuatro del proyecto. Elaboración propia.

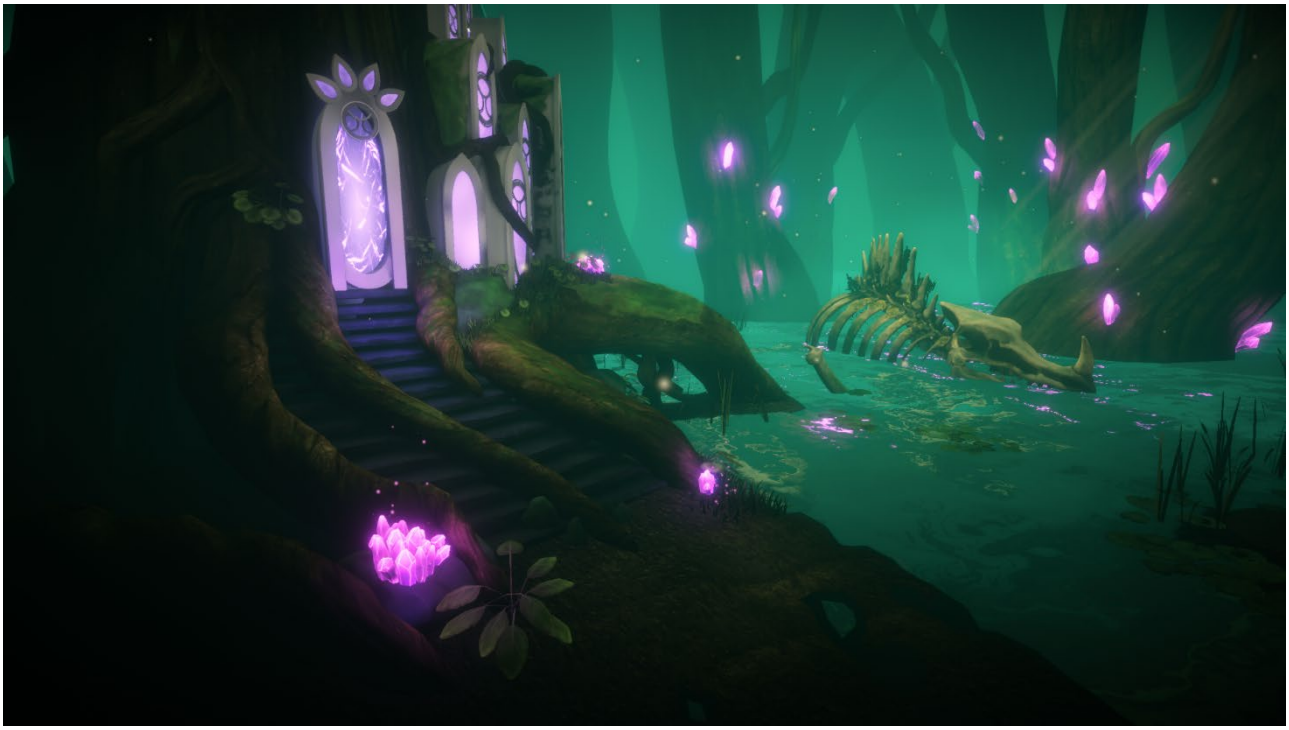

Fig. 51. Render final cinco del proyecto. Elaboración propia.

8. RESULTADOS CONCURSO

Una vez cerrado el concurso de The Rookies, se sumaron un total de 110 participantes de 23 países diferentes. La gran parte de los escenarios mostraban un alto nivel de ejecución ya que la posibilidad de utilizar otros motores en tiempo real como Unreal o adquirir los *assets* y texturas en galerías como Megascans ha permitido a muchos participantes crear escenarios asombrosos.

Lamentablemente, no se ha llegado a obtener ninguna clasificación siendo los ganadores:

1) Primer puesto: Arnaud Claudet de Francia (ver Fig.52).

Fig. 52. Escenario ganador del concurso de Arnaud Claudet. Recuperado de: https://www.therookies.co/entries/9819

2) Segundo puesto: Georg Klein de Alemania (ver Fig.53).

Fig. 53. Escenario subganador del concurso de Georg Klein. Recuperado de: https://www.therookies.co/entries/9198

3) Tercer puesto: Edouard Boudet de Francia (ver Fig.54).

Fig. 54. Escenariodel tercer ganador del concurso de Edouard Boudet.
Recuperado de: https://www.therookies.co/entries/9270

Sin embargo, los jueces del concurso hicieron una lista de menciones honorificas donde se incluían a los participantes cuyos trabajos les había generado una gran impresión. Además, les otorgaron la insignia de *"Excellence Award"* que permiten alabar el trabajo realizado por los participantes. Finalmente, fui premiada con una de estas menciones honorificas.

9. CONCLUSIONES

La elaboración de este trabajo ha sobrepasado las expectativas iniciales permitiendo profundizar en técnicas de optimización que ha mejorado tanto la calidad de los resultados como los tiempos de producción. Es por ello por lo que considero esta experiencia un buen comienzo para perfeccionar los futuros proyectos que permitan desarrollarme en la integración del sector profesional.

Entre las técnicas y soluciones aprendidas quiero destacar la retopología manual, el desarrollo y proyección de mapas de normales y la iluminación dentro de un motor de

videojuegos puesto que, a pesar de que sí conocía el flujo de trabajo de estos procesos, se ha tenido que llevar a cabo una investigación para profundizar y obtener los mejores resultados.

En lo que respecta a los objetivos planteados al inicio del proyecto, considero el cumplimiento de cada uno de ellos ya que se ha logrado interpretar de forma correcta el diseño bidimensional del artista Svetlin Velinov y, sobre todo, se ha conseguido optimizar lo suficiente para que pueda ser implementado en cualquier motor de videojuegos que permita la jugabilidad en plataformas medio-altas.

Las competencias adquiridas durante la formación recibida en el grado de Comunicación Audiovisual han aportado valor en lo referente a la documentación, autoaprendizaje y solución de problemas, además de ayudar a la composición creada para el renderizado, cuyos conocimientos en fotografía e iluminación has sido de especial relevancia.

Haber asumido todo el apartado gráfico de un escenario ha supuesto un reto personal además de una gran carga de trabajo ya que, en el desarrollo de videojuegos, cada equipo cuenta con al menos tres personas por cada especialización. Sin embargo, el haber trabajado estas disciplinas ha supuesto experimentar y comprender mejor la relación entre los diferentes procesos.

Siendo esta la primera vez en enfrentarme al desarrollo de un trabajo académico de esta envergadura, considero que se ajusta a los términos especificados en la rúbrica y valoro esta experiencia de forma positiva ya que me permitirá abordar con más soltura tanto futuros trabajos con mayor complejidad como la futura Tesis de Máster, ya que pretendo continuar ampliando mis conocimientos hasta alcanzar un perfil profesional de artista 3D.

10. BIBLIOGRAFÍA

A. (2020). *Concept Art: ¿qué es y por qué es tan importante?* Arteneo.
Recuperado de: https://www.arteneo.com/blog/3d-blog/concept-art-que-es-por-que-es-importante/

Aguilar, G. [Gabriel Aguiar Prod.]. (2020, 7 abril). *Unity VFX Graph - Magic Orb Effect Tutorial* [Vídeo]. YouTube.
Recuperado de: https://www.youtube.com/watch?v=7bMOhNUA1bI&ab_channel=GabrielAguiarProd.

Aladente, D. (2020). *Proceso completo de modelado de un personaje para videojuego AAA.* (TFG). Universidad Politécnica de Valencia, Grado en Diseño y Tecnologías creativas. Valencia.
Recuperado de: https://riunet.upv.es/bitstream/handle/10251/148487/Alandete%20-%20Proceso%20completo%20de%20modelado%20de%20un%20personaje%20para%20videojuego%20AAA.pdf?sequence=1&isAllowed=y

Alphas | ZBrush Docs. (s. f.). Pixologic.
Recuperado de: http://docs.pixologic.com/user-guide/3d-modeling/sculpting/sculpting-brushes/alphas/

Asins, J. (2015). *Desarrollo 3D de un nivel de videojuegos*. (TFG). Universidad Politécnica de Valencia, Grado en Bellas Artes. Valencia.
Recuperado de: https://riunet.upv.es/bitstream/handle/10251/45846/Asins%20Ferrandiz%20Jaime_TFG_MEMORIA_Scare%20for%20Sale.pdf?sequence=1

Bourykina, Y. (2018, 12 octubre). *handpainted processes*. Artstation.
Recuperado de: https://ybourykina.artstation.com/blog/GQ7Y/handpainted-processes

Trazos. (2021, 7 abril). *Retopología en 3D ¿Qué es y para qué sirve?* Escuela Trazos.
Recuperado de: https://trazos.net/retopologia-en-3d-que-es-y-para-que-sirve/

Introduction to ShaderGraph. (2021, 13 agosto). Unity Learn.
Recuperado de: https://learn.unity.com/tutorial/introduction-to-shader-graph#5f500900edbc2a0022843fb3

Lima, N. [Nelson Lima 3D]. (2018, 2 julio). *Retopologia de un cráneo animal* [Vídeo]. YouTube.
Recuperado de: https://www.youtube.com/watch?v=dFE1XZXmFLs&ab_channel=nelsonlima3D

Marmoset [Marmoset]. (2017, 3 enero). *Getting to Know Toolbag 3 | Ep. 4: Texture Baking* [Vídeo]. YouTube.
Recuperado de: https://www.youtube.com/watch?v=9_2LlmbFBbc&ab_channel=Marmoset

Mayo, D. (2016). *Diseño y modelado de un personaje para videojuegos 3D*. (TFG). Universidad Politécnica de Valencia, Grado en Bellas Artes. Valencia.
Recuperado de: https://riunet.upv.es/bitstream/handle/10251/73675/MAYO%20-%20DISE%C3%91O%20Y%20MODELADO%20DE%20UN%20PERSONAJE%20PARA%20UN%20VIDEOJUEGO%203D.pdf?sequence=1

Monstruos del Diseño. (2021, 23 abril). *Qué es la Escultura Digital 3D.*
Recuperado de:https://monstruosdeldiseno.com/podcast/que-es-escultura-digital-3d

Notari Arambul, A (2018) *Diseño y concept art de un videojuego: escenario.* (Trabajo fin de grado).
Recuperado de: https://riunet.upv.es/bitstream/handle/10251/108542/NOTARI%20%20Dise%C3%B1o%20y%20concept%20art%20de%20un%20videojuego%20original%3A%20escenarios.pdf?sequence=1&isAllowed=y#:~:text=1.4.,LAS%20ESPECIALIDADES%20DEL&text=Dentro%20del%20concept%20art%20hay,sus%20propios%20objetivos%20y%20peculiaridades.

Poly.Clay. (s. f.). Decimation Master. Guía Poly-Clay.
Recuperado de:https://guiapolyclay.jimdofree.com/preparaci%C3%B3n-del-modelo-3d-para-un-game-engine/decimation-master/

Selin, E. (2021, 27 abril). *10 Different types of 3D modeling techniques. Artisticrender*.Com. Recuperado de: https://artisticrender.com/10-different-types-of-3d-modeling-techniques/

Texturas 3D. (2020, 13 mayo). *¿Qué son las texturas PBR y cómo crearlas desde fotogrametría?* Recuperado de: https://www.texturas3d.com/fotogrametria/texturas-pbr/

The Rookies. (2021). *About our community and mission.* The rookies. Recuperado de https://www.therookies.co/about

Pluralsight. (2017, 1 junio). *Love Them or Hate Them, They're Essential to Know*. Recuperado de: https://www.pluralsight.com/blog/film-games/understanding-uvs-love-them-or-hate-them-theyre-essential-to-know

U. Technologies. (2019). *Creating a Global Post Processing Volume.* Unity Learn. Recuperado de: https://learn.unity.com/tutorial/creating-a-global-post-processing-volume#5d0b3fc0edbc2a00205f34cb

U. Technologies. (2018). *Iluminación baked - Unity Manual*. Unity. Recuperado de: https://docs.unity3d.com/es/2018.4/Manual/LightMode-Baked.html

U. Technologies. (2018). *Tipos de luz - Unity Manual.* Unity. Recuperado de:https://docs.unity3d.com/es/2018.4/Manual/Lighting.html

U. Technologies. (2018). *Utilizando Sistemas de Partículas en Unity* - Unity Manual. Unity. Recuperado de: https://docs.unity3d.com/es/2018.4/Manual/PartSysUsage.html

World of Real-time Discoveries. (s. f.). Recuperado de: https://www.therookies.co/contests/groups/world-of-discoveries

11. ANEXO

Galán Puerto, M. (2021). Cinemática final del proyecto: https://floridauniversitariamy.sharepoint.com/:v:/g/personal/magapu_floridauniversitaria_es/EUzYu¡DEHVRRGvqqPzg1x8UMB9fMt3KuxftpUELhId5bS_Q?e=TqQgfB