



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Facultad de Bellas Artes

Aether II. Desarrollo de un personaje y entorno en 3D

Trabajo Fin de Grado

Grado en Diseño y Tecnologías Creativas

AUTOR/A: Balica , George Alexandru

Tutor/a: Abad Cerdá, Francisco José

CURSO ACADÉMICO: 2021/2022

RESUMEN Y PALABRAS CLAVE

RESUMEN

Este trabajo consiste en el desarrollo de un personaje humano y otro animal, los cuales se integran en una composición que simula un entorno. Se realiza mediante modelados 3D empleando diversas herramientas digitales y se ejecuta en el motor de videojuegos *Unreal Engine 5*. Se parte de un trabajo previo de concept comprendido en el TFG *AETHER I*, en el cual se establece una estética general y diseño base de los elementos a realizar en este proyecto. Para la etapa de producción del proyecto se trabaja el modelado de ambos personajes mediante diversas técnicas empleando programas como *Blender* y *Marvelous Designer*. Tras esto se trabajan las texturas, materiales y riggeados de dichos personajes para que puedan ejecutar diversas poses. Por último, se realiza el montaje de un escenario mediante assets en el motor de videojuegos *Unreal Engine 5*, al cual se añaden efectos de luz y otras características que implementa este motor. La idea de este proyecto es presentar todos los elementos desarrollados en un encuadre visualmente atractivo el cual muestre la cohesión del conjunto. El objetivo detrás de todo esto es sacar adelante un proyecto multidisciplinar que abarque diferentes campos de la industria del 3D, sirviendo así como un ejercicio de portfolio versátil y como demostración de todo lo aprendido a lo largo de la carrera.

SUMMARY

This project consists of the development of a human and an animal character, which are integrated into a composition that simulates an environment. It is done through 3D modeling using various digital tools and runs on the "Unreal Engine 5" video game engine. It starts from a previous concept work described in a TFG carried out by another student, in which a general aesthetic and a basic design of the elements to be carried out are established. In the production stage of the project, the modeling of both characters is worked on through various techniques using programs such as Blender and Marvelous Designer. Subsequently, the textures, materials and rigging of the characters are worked on so that they can execute various poses. Finally, a scenario is assembled using assets in the video game engine Unreal Engine 5, to which light effects and other features implemented by this engine are added. The main objective of this project is to present all the elements developed in a visually attractive frame that shows the cohesion of the whole. Another objective is to carry out a multidisciplinary project that covers different fields of the 3D industry, thus serving as a versatile portfolio exercise and as a demonstration of everything learned throughout the degree.

PALABRAS CLAVE

Blender; Unreal Engine; modelado 3D; escenario; personaje

KEY WORDS

Blender; Unreal Engine; modelado 3D; scenery; character

CONTRATO DE ORIGINALIDAD

El presente documento ha sido realizado completamente por el firmante; es original y no ha sido entregado como otro trabajo académico previo, y todo el material tomado de otras fuentes ha sido citado correctamente.

AGRADECIMIENTOS

Gracias a Azahar Ferrándiz Almagro, por hacer posible este proyecto y por la buena comunicación que ha habido a lo largo del proceso de trabajo.

Gracias también a Francisco José Abad Cerdá, tutor de este TFG, por las correcciones que han mejorado el aspecto formal de este trabajo.

Por último gracias a mi familia, amigos y profesorado, quienes han ofrecido su apoyo y han despertado el interés por este tipo de proyectos.

ÍNDICE

1. INTRODUCCIÓN	7
1.1. JUSTIFICACIÓN	7
1.2. OBJETIVOS	7
1.1.1. OBJETIVOS PRINCIPALES	7
1.1.2. OBJETIVOS SECUNDARIOS	7
1.3. METODOLOGÍA	8
2. DESARROLLO	9
2.1. BRIEFING	9
2.2. REFERENTES PROFESIONALES	10
2.2.1. AETHER I	10
2.2.2. REFERENTES DE PERSONAJES	10
2.2.2.1. <i>MINOH KIM.</i>	10
2.2.2.2. <i>THIBAUT GRANET.</i>	11
2.2.2.3. <i>YEKATERINA BOURYKINA.</i>	11
2.2.3. REFERENTES DE ESCENARIOS	11
2.2.3.1. <i>SIMON FUCHS.</i>	11
2.2.3.2. <i>BRANDON MARTYNOWICZ.</i>	12
2.3. DESCRIPCIÓN DEL PROCESO DE DESARROLLO	12
2.3.1. GALIA	13
2.3.1.1. <i>MODELADO</i>	13
2.3.1.2. <i>TEXTURIZADO</i>	14
2.3.1.3. <i>POSE</i>	15

2.3.2. SAKE	15
2.3.2.1. <i>MODELADO</i>	15
2.3.2.2. <i>TEXTURIZADO</i>	16
2.3.2.3. <i>POSE</i>	17
2.3.3. ESCENARIO	18
2.3.3.1. <i>MODELADO</i>	18
2.3.3.2. <i>TEXTURIZADO</i>	18
2.3.4. UNREAL ENGINE 5	19
2.3.4.1. <i>IMPORTACIÓN DE RECURSOS</i>	20
2.3.4.2. <i>LUCES</i>	20
2.3.4.3. <i>RENDERIZADO</i>	21
2.4. RESULTADO	22
2.5. DIFUSIÓN	22
3. CONCLUSIONES	28
4. BIBLIOGRAFÍA	29
5. ÍNDICE DE FIGURAS	31

1. INTRODUCCIÓN

Este proyecto se centra en la creación de una serie de personajes, elementos y entorno que en conjunto conforman una imagen cohesionada. La idea surge de productos vistos en plataformas como la Workshop de Steam, donde se crean y comparten fondos de pantalla animados tanto en 3D como en 2D; o en Artstation, donde artistas 3D y 2D presentan sus proyectos personales a modo de “escaparate” donde observamos el resultado final y las etapas de desarrollo de este.

1.1. JUSTIFICACIÓN

Un proyecto de estas características permite llevar a la práctica técnicas aprendidas durante la carrera y otras que se han aprendido de manera independiente. De esta manera, se crea un trabajo de portfolio personal que sirve para familiarizarse con herramientas o procesos de trabajo que se acerquen lo máximo posible a lo que supondría un proyecto profesional del ámbito del 3D en general.

1.2. OBJETIVOS

1.1.1. *Objetivos Principales*

La prioridad a la hora de realizar este proyecto es ampliar el portfolio, ya que es el medio principal para demostrar el progreso y las habilidades personales en esta industria. Por ello en este trabajo, si bien es cierto que no se llega a profundizar en exceso en cada técnica empleada, se trata de trabajar diversos apartados para así descubrir aquellos en los que se necesite mejorar.

1.1.2. *Objetivos secundarios*

De manera secundaria, nos encontramos con una serie de factores que influyen tanto en la etapa de desarrollo como en el aprendizaje obtenido del mismo.

En primer lugar, se trata de aprender las técnicas adecuadas para cada proceso empleado, las cuales se utilizan en el ámbito profesional para optimizar tanto recursos como tiempo.

Concretamente, aprender a tratar de manera correcta los diversos materiales, conseguir trabajar de manera correcta y eficiente los UVs de las mallas, experimentar con herramientas nuevas, crear modelos con una estética estilizada e implementar todo esto en un motor de videojuegos de última generación.

En segundo lugar, explorar el proceso creativo que engloba la creación de un universo nuevo que, si bien es cierto que se basa en una serie de re-

ferentes, da la posibilidad de experimentar con diversos elementos, como la creación de vestuario y entornos reconocibles.

En tercer lugar, al basarse en un trabajo de diseño existente, practicar la habilidad de plasmar las ilustraciones conceptuales en 2D de otra persona en una puesta en escena tridimensional, conservando los elementos que hacen reconocibles tanto al personaje como al entorno.

1.3. METODOLOGÍA

A lo largo de la etapa de desarrollo se han empleado diversas técnicas, divididas entre el desarrollo de los personajes y el escenario. En el caso del personaje humano, se parte de una malla base a la cual se le ajustan las proporciones en *Blender*, se le dota de las facciones faciales correspondientes al personaje previamente diseñado, y se le añaden elementos como el pelo, pestañas y ojos. Tras esto, se crea su vestuario en *Marvelous Designer*, programa que hace uso de técnicas de patronaje para confeccionar vestimentas que se ajustan a la malla de nuestro personaje con sus correspondientes físicas. Al volver a *Blender*, se modelan los *props* restantes del personaje y se prepara todo para la etapa de *rig*. En este punto, se hace uso de *Mixamo*. Por último, pasamos a añadir las texturas al personaje mediante el uso de diversos materiales, mapas de normales y desplazamiento, pintando sobre los *UVs* previamente dispuestos para dicha tarea.

Para el personaje animal, que es un gato, partimos desde cero creando una malla que se aproxime lo máximo posible a la forma final del felino. Tras esto, se subdivide la malla y se esculpe su pelaje haciendo uso de *alphas* para darle forma. Se modelan por separado los bigotes, ojos y almohadillas de las patas y se textura al igual que con el personaje anterior. En este caso, para el *rig* del gato, hacemos uso de la extensión de *Blender* llamada *Rigify*, para obtener el esqueleto de un felino que debemos adaptar a nuestra malla y posteriormente anclar los elementos de la malla a cada hueso de manera manual.

Por último, el escenario consiste en el modelado de diversas piezas que nos permitan montar estructuras variadas y que mantienen cohesión entre sí. Se les asignan diferentes opciones de materiales a cada una y se ensambla todo en *Unreal Engine 5*.

- *Props*: Término utilizado para definir elementos que acompañan al personaje o entorno.
- *Rig*: Proceso mediante el cual se dota de esqueleto a un personaje para su posterior animación.
- *UVs*: Cada una de las caras de una malla en 3D que se plasma sobre una imagen para agregarle textura.
- *Alphas*: Imágen monocromática semitransparente que determina la forma y textura de un pincel digital.

Durante el ensamblaje se añaden luces y efectos que dotan a la escena de una mayor complejidad y credibilidad. Cabe destacar que la reducción del nivel de detalle de las mallas se ha descuidado de manera intencionada para poner en uso una nueva herramienta de *Unreal Engine 5* llamada *Nanite*, la cual recalcula la geometría de objetos estáticos a partir de su resolución en pantalla en tiempo real. También se utiliza *Lumen*, una herramienta que permite un comportamiento de la luz más complejo y en tiempo real haciendo uso del *trazado de rayos*.

Tras todo esto se realizan varios renders que dejen ver todos los elementos, tanto por separado como en conjunto.

2. DESARROLLO

2.1. BRIEFING

Para llevar a cabo este proyecto se toman como base los diseños realizados en *Aether I*, en los cuales se han desarrollado los rasgos físicos y vestimentas de Galia, el personaje principal, y de Sake, el gato que la acompaña. Asimismo, también define el escenario y encuadre realizados para la puesta en escena de dichos personajes y el diseño del entorno.

Partiendo de esto, el objetivo es plasmar el arte conceptual anteriormente mencionado. Esto se consigue modelando en 3D por separado cada uno de los elementos que componen este trabajo, utilizando las técnicas más convenientes para cada uno. Tras ello, se crean los materiales correspondientes para aplicar texturas a dichos modelos y se realiza el rig de los personajes, para permitir realizar diversas poses para la posterior puesta en escena.

Una vez están tanto el escenario como los personajes listos, se realiza la migración desde *Blender*, programa utilizado para la mayor parte del proyecto, a *Unreal Engine 5*.

Se pretende conseguir una imagen compuesta que plasme la ilustración realizada en la etapa de concept y que presente todos sus elementos de manera coherente.

2.2. REFERENTES PROFESIONALES

Uno de los aspectos principales que se tiene en cuenta a la hora de realizar este proyecto es que haya una cohesión estilística entre los elementos a realizar. Por ello, tomamos como referentes una serie de productos que, pese a pertenecer a géneros diferentes, mantienen una relación de estilo entre ellos. Si bien es cierto que el objetivo no es plasmar al pie de la letra la estética de dichos referentes, se han tomado como inspiración para desarrollar diversos elementos de la historia y ambientación de Aether. Además, se tiene en cuenta el trabajo previo de sus artistas, así como artistas pertenecientes a proyectos similares.

2.2.1. Aether I

Este proyecto nace como una continuación de *Aether I*, realizado por Azahar Ferrándiz Almagro. Ella es una alumna de Bellas Artes que ha realizado dicho trabajo para su TFG este curso. En él se ha elaborado la historia de *Aether*, una ciudad futurista que emplea un elemento alquímico del mismo nombre como base de sus avances tecnológicos, y la de sus personajes, los cuales aprovechan restos de tecnología para realizar inventos. Se han elaborado los diseños de sus personajes y escenario como trabajo de concept art que queda reflejado en un libro de arte anexo a *Aether I*.

Dada la naturaleza de ambos proyectos, se ha considerado esencial la buena comunicación entre ambos, ya que esto facilita la transmisión de ideas, feedback y organización. La dinámica seguida para llevar a cabo este proyecto ha sido la de trabajar simultáneamente a medida que se han ido haciendo avances en cada parte del trabajo, pudiendo avanzar ambos a la vez en cada uno de nuestros proyectos.

Ya que *Aether II* continúa el trabajo realizado en *Aether I*, se ha tomado como principal referente, y se trata de plasmar la visión general del proyecto original.

2.2.2. Referentes de personajes

2.2.2.1. Minoh Kim. Se trata de uno de los responsables del desarrollo de personajes en *Riot Games*. Actualmente reside y trabaja en San José, California. Ha sido uno de los artistas responsables de modelar diversos personajes para el videojuego *Valorant*, de la misma compañía. En sus diseños destaca el uso de elementos futuristas y una estética estilizada, que genera siluetas reconocibles.

Las proporciones de sus personajes y el acabado de ciertas texturas son algunos de los principales elementos que se han tenido en cuenta a la hora de realizar *Aether II*.



Figura 1. Concept final de *Aether I*.
Por Azahar Ferrándiz Almagro.



Figura 2. Giro de Viper, de *Valorant*.
Por Minoh Kim.



Figura 3. Giro de Jinx, de Arcane.
Por Thibaut Granet.



Figura 4. Hoja de personaje de Caitlyn,
de League of Legends.
Por Yekaterina Bourykina.



Figura 5. Toronto, mapa de Overwatch.
Por Simon Fuchs.

2.2.2.2. Thibaut Granet. Este artista francés, que actualmente reside en París, ha trabajado para *Fortiche*, la empresa encargada de llevar a cabo la producción de *Arcane*, serie de *Riot Games*. Este artista trabaja principalmente con el modelado y texturizado de personajes, y ha sido el responsable de llevar a cabo parte del desarrollo de uno de los personajes principales de la serie. Anteriormente también ha trabajado en la animación de *Rise* y de *K/DA*, de la misma compañía.

Dado el estilo general y la historia de la serie, al cual se asemeja en cierto modo *Aether*, se ha tomado como referencia sobre todo para elementos de vestuario, que complementan la narrativa y ambientación que se quieren transmitir.

2.2.2.3. Yekaterina Bourykina. Al igual que en los casos anteriores, esta artista es una de las encargadas del diseño de personajes en *Riot Games*. Su trabajo ha consistido en la realización de diversos personajes y skins para *League of Legends*, y actualmente está trabajando como jefa del departamento de personajes en un videojuego MMO de *Riot Games*, de nombre aún desconocido. Otras aportaciones de esta artista al mundo de los videojuegos han sido en *World of Warcraft*, de *Blizzard Games*, y en *Aphelion*, de *Lunatic Studios*. Una de sus principales virtudes es la gran versatilidad entre diseños que muestran sus trabajos, haciendo evidente que es capaz de plasmar ideas muy diferentes entre sí y obtener buenos resultados.

Al igual que en el caso anterior, se han tomado como inspiración los props y el vestuario que acompañan a sus personajes.

2.2.3. Referentes de escenarios

2.2.3.1. Simon Fuchs. Un artista residente en Seattle con más de 17 años de experiencia en la industria de los videojuegos. Ha trabajado en múltiples títulos conocidos, tales como: *Half-Life*, *Overwatch*, *Starcraft II*, etc. Actualmente trabaja como artista 3D en *Valve* y comparte sus conocimientos sobre el oficio online mediante cursos de pago en la página *Gumroad*. Su especialización es el modelado y texturizado de props y escenarios, y es por esto mismo por lo que es tomado como referente para este proyecto.

Para *Aether* es especialmente interesante la etapa en la que este diseñador ha trabajado para *Blizzard Entertainment*, concretamente a la hora de realizar diversos mapas del videojuego *Overwatch*, donde se mezclan elementos futuristas con otros antiguos bajo una estética estilizada.



Figura 6. Split, mapa de Valorant.
Por Brandon Martynowicz.

2.2.3.2. Brandon Martynowicz. Este es uno de los artistas principales en el desarrollo escenarios para *Riot Games*. Actualmente reside en Santa Mónica, California. Entre sus trabajos más destacados están el modelado de props para el videojuego *Call of Duty: Black Ops III* y el diseño de algunos escenarios para del videojuego *Valorant*.

Su estilo muestra versatilidad a la hora de adaptarse a la estética particular del proyecto en el que esté trabajando. Esto, junto con su habilidad para crear entornos coherentes y estilizados lo hacen un buen referente para tratar de crear la puesta en escena de Aether.

Otros artistas similares, que trabajan también para *Riot Games*, son Anessa Silzer y Devon Fay.

2.3. DESCRIPCIÓN DEL PROCESO DE DESARROLLO

Tras haber analizado el trabajo realizado en *Aether I*, ha sido necesaria una planificación previa del flujo de trabajo para poder llevar a cabo un desarrollo del proyecto ordenado y limpio.

En primer lugar, ha sido muy importante la gestión de carpetas y archivos pertenecientes a este proyecto. Esto se ha hecho para facilitar el acceso a cada elemento, la migración de los archivos de un programa a otro y el tener localizadas cada una de las versiones a medida que se ha ido avanzando.

En segundo lugar, por seguridad, se han ido haciendo una serie de copias de seguridad, que se han ido subiendo a la nube. Esto se ha hecho tanto para evitar la pérdida del progreso en caso de que hubiese algún problema de hardware, como para poder regresar a una versión anterior en caso de querer deshacer algunos cambios.

Por último, se ha hecho una selección previa del software a utilizar, ya que en caso de no disponer de los programas o conocimientos necesarios para realizar algunas etapas del desarrollo habrían de buscarse alternativas. Los programas que se han acabado utilizando para este proyecto han sido *Blender* (versión 3.1.2), *Marvelous Designer 10*, *Substance Painter*, *Photoshop 2022* y *Unreal Engine 5*. Adicionalmente se han utilizado herramientas en línea tales como *Quixel* y *Mixamo*.

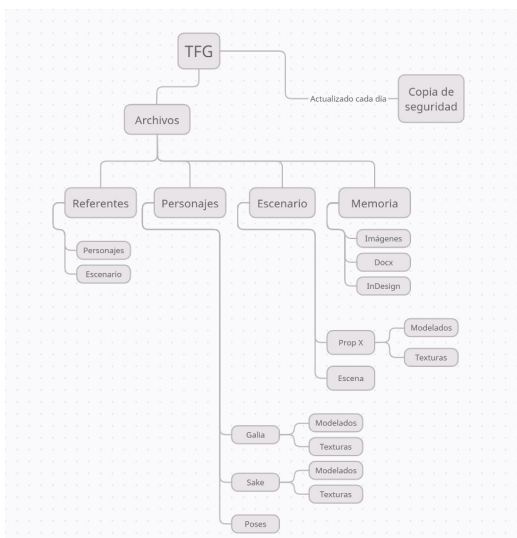


Figura 7. Esquema de organización de carpetas.



Figura 8. Diseño de Galia.
Por Azahar Ferrándiz Almagro.

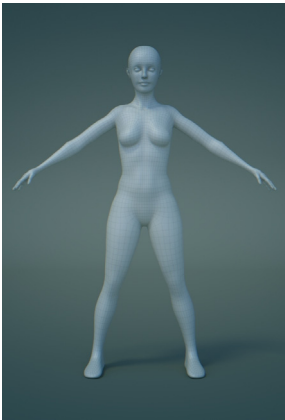


Figura 9. Malla base femenina.
Por TheEmptyRoom.

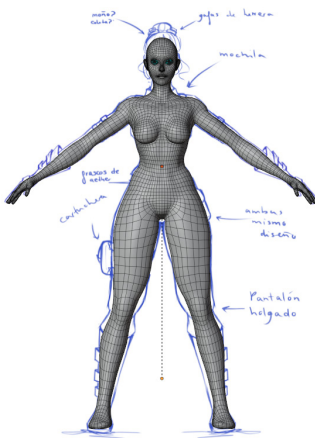


Figura 10. Ajuste de proporciones de la malla según los bocetos.



Figura 11. Estructura base del pelo basada en puntos que indican la dirección.

2.3.1. Galia

Se trata de un personaje femenino, de complexión atlética, la cual se dedica a ir buscando restos de tecnología para sus inventos y experimentos. Sus vestimenta la compone elementos robustos como botas de seguridad, guantes de cuero y gafas de protección. Asimismo, también lleva una mochila y diversas correas y compartimentos para llevar sus materiales. Sus ropas facilitan el movimiento y lleva su cabello rojo recogido en un moño para que no le estorbe mientras trabaja.

2.3.1.1. Modelado. Al tratarse de un personaje humanoide, se ha partido de una malla base desde la que empezar a trabajar. En este caso, se ha recurrido a descargar una malla base gratuita, ya que ahorra mucho tiempo de desarrollo y evita posibles problemas de topología para más adelante. Se trata de una malla base descargada de la página *TurboSquid*, realizada por el usuario *TheEmptyRoom*.

A continuación, se modificaron las proporciones de la malla base en *Blender* para que coincidan con las del diseño realizado en *Aether I*. Galia se caracteriza por tener unas piernas largas y robustas, un torso proporcionalmente más corto y unos brazos finos. Sus facciones faciales son alargadas, finas y cuenta con una cabeza algo más pequeña proporcionalmente que la del modelo base que se ha utilizado. Su proporción general es de aproximadamente siete cabezas y media de altura total.

A continuación, se comenzó a modelar el pelo en torno a la cabeza. No se buscaba una estética en la que se distinga cada pelo, sino que se buscaba algo más estilizado. Cada mechón conforma una malla que se puede asemejar a tiras de arcilla con cierta textura que lo hacen creíble.

El modelado de cada mechón parte de un círculo en *Blender*, que se ha subdividido y modificado para generar curvas y vértices a lo largo de la circunferencia. Esta forma que se ha generado será la que determine el perfil que tendrá el mechón de pelo al que se le asigne. Se han realizado variaciones en cada parte del peinado. Se crea una línea de vértices a la cual se le aplica el modificador *geometry* y se le asigna la forma anteriormente creada. Se puede modificar el tamaño de la geometría de cada vértice utilizando el comando *Ctrl+A*. Se coloca la línea de vértices en la posición deseada sobre la cabeza y se le añade el modificador *mirror* en caso de ser un mechón simétrico. Una vez construido el peinado al completo se le aplican los modificadores y se unifican las mallas.

Se agregan también las pestañas y las cejas mediante planos a los cuales se les aplican los modificadores *solidify* y *subdivision surface*.

- *Geometry*: Modificador que permite que una curva adopte el perfil de una la forma que se le especifique.
- *Solidify*: Modificador que dota de volumen al plano que se le aplique.
- *Subdivision Surface*: Modificador que suaviza las aristas de una geometría mediante subdivisión.

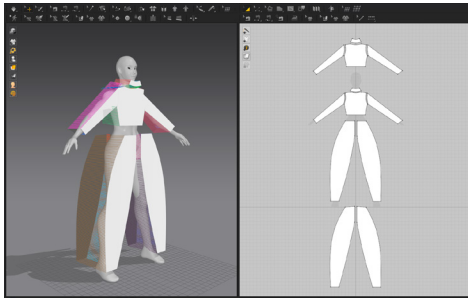


Figura 12. Patronaje del vestuario junto a la orientación de las costuras.



Figura 13. Mallado definitivo de Galia, su vestuario y props.

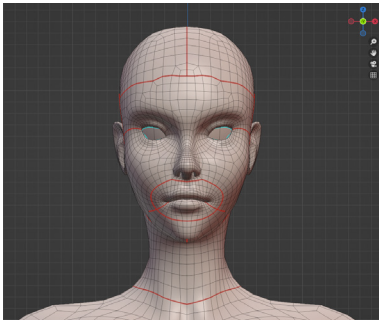


Figura 14. Segmentación del rostro de Galia para trabajar los UVs.

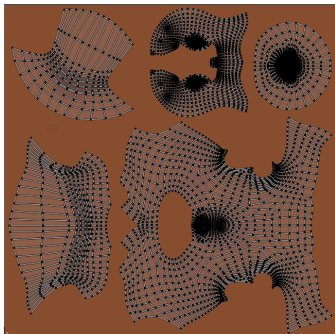


Figura 15. UVs del cuerpo y rostro de Galia sobre su textura.



Figura 16. Texturas definitivas aplicadas a la malla de Galia.

Una vez terminada la primera etapa en *Blender*, se exportó la malla de Galia a *Marvelous Designer* para confeccionar su vestimenta. En este programa hubo que diseñar las formas de los retales que conforman la ropa indicando la ubicación y orientación de las costuras mediante el uso del patronaje. En este caso la ropa consistía en un top de manga larga y cuello alto, y unos pantalones un poco holgados.

Para completar la vestimenta de Galia, se ha modelado el resto de accesorios que la acompañan en *Blender* sobre el modelo con la ropa ya puesta. Estos elementos son unas gafas, guantes, botas, diversas correas, una mochila, unos frascos y un par de bolsas que lleva en la cintura y en la pierna.

Una vez completado el modelado de todos los elementos, se han aplicado los modificadores, se han unido las mallas correspondientes, y se les ha aplicado la jerarquía correspondiente mediante la herramienta *parent*.

2.3.1.2. Texturizado. Como el vestuario de este personaje está compuesto de objetos de materiales muy diferentes, ha sido necesario un trabajo previo de preparación para poder empezar a aplicar texturas. Este trabajo ha consistido en ir separando secciones de la malla para indicar las incisiones a la hora de desplegar los UVs. Una vez realizado este proceso para cada objeto, se empezaron a aplicar directamente desde *Blender* las texturas más sencillas, y que consistían en colores planos con alguna variación hecha a pincel en el modo *texture paint*. Dichos materiales han sido: la piel, los ojos, el pelo, el metal de los accesorios y el cristal de los frascos y las gafas.

Para los materiales más complejos se ha hecho una búsqueda de imágenes con las texturas correspondientes para que coincidan con los materiales deseados. Dichos materiales han sido dos tipos de telas para la ropa, y una textura de cuero. Se han generado sus correspondientes mapas de normales y desplazamiento con el uso de *Photoshop*. También se han modificado ligeramente las tonalidades y se han hecho algunas variaciones de color para el cuero.

De cara a optimizar el uso de materiales en el proyecto se ha utilizado una técnica llamada *trim sheet*, la cual permite componer varios materiales en una sola imagen. La variación de materiales según la parte de la malla que nos interese se consigue reubicando y escalando los UVs correspondientes a dicha parte de la malla. Lo que se consigue de esta manera es aligerar el peso del proyecto y hacer más rápido el proceso de aplicar texturas.

- *Parent*: función que establece una relación de parentesco entre dos objetos 3D en la que uno de ellos, el padre, influye al otro, el hijo.



Figura 17. Estructura de los huesos del rig de Galia.
Generado con Mixamo.



Figura 18. Diseño de Sake.
Por Azahar Ferrándiz Almagro.

2.3.1.3. Pose. Tradicionalmente, y por motivos de funcionalidad, los personajes modelados en 3D se les diseña en *T-pose*. Esto se debe a que es una posición que facilita tanto el modelado como el posterior riggeado, minimizando así deformaciones indeseadas en la malla y un flujo de trabajo incómodo.

Habiendo modelado a Galia siguiendo este tipo de pose, llegó el momento de asignarle un esqueleto para hacer posible que se mueva. Si bien es cierto que el objetivo de este trabajo no es animar al personaje, el riggeado del mismo permite colocarlo en diversas poses para cualquier renderizado que se quiera obtener.

En este caso se ha hecho uso de *Mixamo* para realizar el rig de este personaje. El proceso consiste en subir a la página el modelo humanoide correspondiente, marcar ciertos puntos clave en la malla y esperar a ser procesado. La página cuenta también con diversas animaciones de stock que se pueden previsualizar antes de descargarlas, pero en este caso sólo interesaba descargar el personaje con su esqueleto. Cabe destacar que para proyectos que requieran mayor complejidad a la hora del movimiento del personaje es recomendable realizar el rig de manera manual, ya que se consiguen resultados más precisos, aunque alarga considerablemente el tiempo de desarrollo.

A continuación, se han ido haciendo pruebas de diversas poses, que tenían como objetivo presentar la versatilidad del personaje. Finalmente, y de cara a la puesta en escena final, se ha optado por una pose en la que Galia parece estar andando, con la cabeza girada ligeramente hacia su izquierda.

2.3.2. Sake

Sake es un gato gris que acompaña a Galia en sus aventuras. Sus ojos son verdes, es ligeramente rechoncho y tiene un pelaje de longitud media y un poco revuelto. Es un felino curioso y que, pese a no separarse normalmente de Galia, sigue conservando su espíritu de gato callejero.

2.3.2.1. Modelado. Al igual que con Galia, el modelado de Sake tenía que partir de una malla de un gato desde la cual empezar a plasmar las características del personaje. Este caso fue un poco más complicado, debido a que, al tratarse de un animal cuadrúpedo, las mallas gratuitas eran inferiores en calidad y no se correspondían con lo que se buscaba.

Aun así, se decidió realizar un primer prototipo utilizando algunas mallas gratuitas, las cuales fueron rápidamente desechadas debido a fallos de topología, acabados por debajo de las expectativas y proporciones que distaban mucho del concepto que se tenía de Sake.

- *T-pose*: Término que hace referencia a la pose que adopta una figura humanoide al extender los brazos hacia los lados y con las piernas ligeramente separadas.

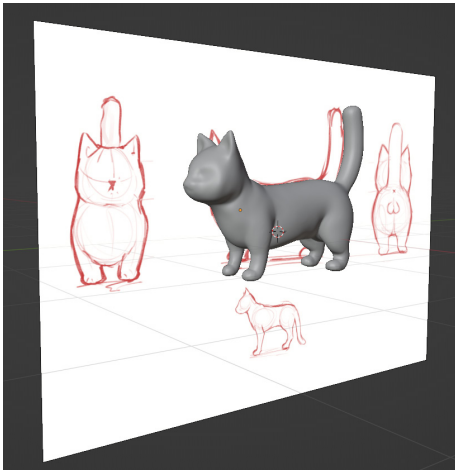


Figura 19. Segunda versión de Sake con los primeros bocetos de fondo.

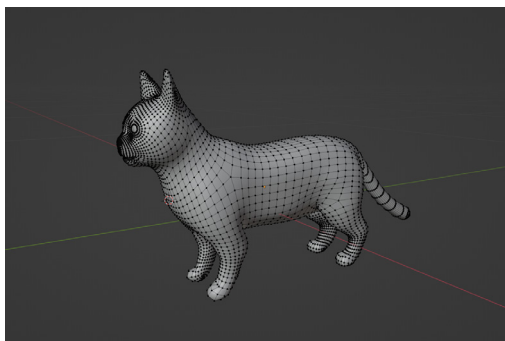


Figura 20. Mallado base definitivo de Sake.



Figura 21. Pelaje de Sake modelado con Alphas.

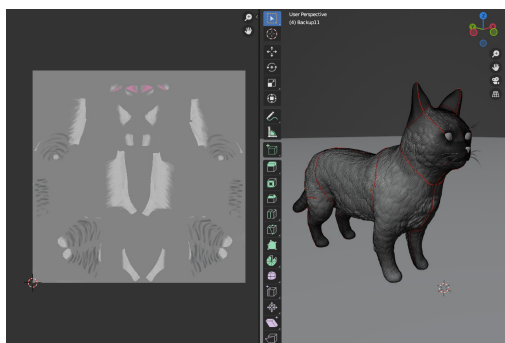


Figura 22. Segmentación de los UVs de Sake junto a su textura.

Las siguientes pruebas una técnica para generar personajes simples. Esta técnica consiste en generar un esqueleto mediante la extrusión de puntos, a los que se les aplican los modificadores de *skin*, *subdivision surface* y *mirror*. De esta manera, se obtenía una malla que posteriormente en el modo esculpir de Blender se podía ir acercando al resultado deseado. Pese a acercarse más al objetivo que se buscaba, esta versión también fue desechada debido a fallos de topología en la malla. Esto sucedía sobre todo en las zonas cercanas a las articulaciones, ya que con esta técnica no se tomaba en cuenta la anatomía de un gato a la hora de generar la malla.

Por último, la técnica que conseguido tener mejores resultado ha sido tomar como referencia mallas de gatos online de pago, las cuales tenían un acabado considerablemente mejor a las gratuitas. Partiendo de la previsualización de este tipo de mallas se ha ido modelando manualmente los polígonos que forman a Sake, teniendo en cuenta su anatomía y acercando el modelo lo más posible al definitivo. Sin duda esta ha sido la etapa más tediosa del proyecto, ya que ha sido la que más tiempo ha llevado para llegar los resultados esperados.

Teniendo ya la malla base de Sake, se procedió a la etapa de detallado del gato. Dicha etapa ha consistido en el modelado de pequeños elementos como los ojos, bigotes, nariz y almohadillas de las patas. A la malla principal se le han aplicado varias subdivisiones y se ha ido esculpiendo para tener una forma más orgánica.

Por último, se ha esculpido el pelaje del gato mediante el uso de *alphas*. Esta técnica consiste en el uso de imágenes a modo de pincel para modificar la superficie de la malla. Este proceso actúa de manera similar a como lo haría un mapa de desplazamiento, con la diferencia de que es necesario aplicar a la malla un modificador de multiresolución, aumentando así el conteo de polígonos a cambio de mayor precisión y definición de las formas.

2.3.2.2. Texturizado. Al tratarse de un personaje sin complementos ni ropa, el proceso de texturizado ha resultado ser considerablemente más rápido que el de Galia, ya que simplemente había que crear texturas para los ojos y el pelaje.

Al igual que con Galia, se comenzó haciendo los cortes necesarios en la malla para aislar diferentes partes y haciendo así que los *UVs* fueran más fáciles de manejar. Las partes que se obtuvieron fueron las patas, el torso, la cola, la cabeza y las orejas.

- *Skin*: Modificador que dota de volumen a la estructura de puntos al que se aplique
- *Mirror*: Modificador que genera efecto espejo de la malla a partir de su origen.



Figura 23. Texturas definitivas aplicadas a la malla de Sake.

Una vez hecho esto, se aplicó color al gato mediante la herramienta *texture paint*, de manera que quedase lo más parecido posible a los concepts previos del gato. El pelaje consistía en una base gris, con el vientre y parte del cuello blanco y unas rayas atigradas más oscuras en las patas y la cola. Los ojos forman un degradado de verdes oscuros y claros y el resto de elementos, como los bigotes y la nariz, están hechos con una textura plana blanca y rosa respectivamente.

Para hacer posible la migración de un programa a otro se realizó el bake de las texturas, de manera que quedasen plasmadas en una imagen y no como nodos en el propio *Blender*.

2.3.2.3. Pose. Al tratarse de un animal cuadrúpedo, no era posible utilizar la misma herramienta para riggear a Sake que la que hemos utilizado para Galia. Esta vez ha sido necesario emplear la herramienta *Rigify*, un plugin de *Blender* que permite la creación tanto de esqueletos humanoides como de algunos animales.

Lo primero que se ha tenido que hacer es generar un esqueleto de gato empleando la herramienta anteriormente mencionada. Este esqueleto se coloca sobre la malla de Sake de manera que ambas estén en yuxtaposición. Tras esto, se activa el modo *X-ray* para ver a través de la malla. Se reposiciona cada uno de los huesos para ocupar el espacio que sería anatómicamente correcto dentro de la malla. Se selecciona el esqueleto y en el panel de herramientas de *Rigify* pulsa la opción de generar rig. Esto dará lugar a los controles de movimiento del gato, que se anclarán a la malla haciendo visibles los huesos que los componen. Seleccionando la malla y, en modo pose, se pulsa *Ctrl+P* para vincular los huesos a la malla con pesos automáticos. Tras esto ha sido necesario anclar los ojos, los bigotes, la nariz y las almohadillas a los huesos correspondientes, ya que se trata de mallas independientes.

Los resultados de esta técnica han sido más precisos y complejos que los obtenidos con *Mixamo*, ya que permite un control total sobre la ubicación de cada hueso y articulación. Adicionalmente, este método genera también huesos que permiten trabajar expresiones faciales.

Por último, se procedió a experimentar con diversas poses para Sake, al igual que con Galia. Se decidió hacer que el gato se encuentre caminando hacia adelante como si estuviera investigando. De esta manera se trata de reflejar su carácter curioso y aventurero.

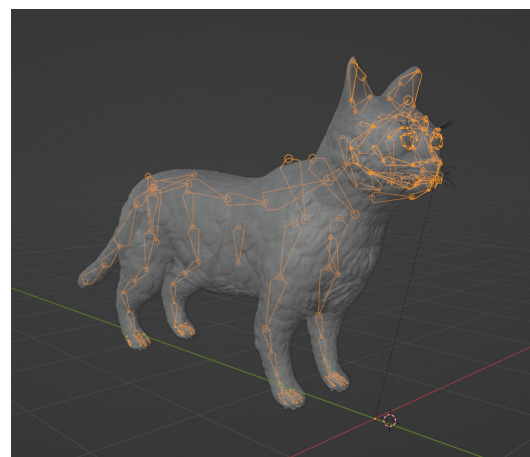


Figura 24. Estructura de los huesos del rig de Sake. Generado con *Rigify* en *Blender*.

- *X-ray*: Modo en el que la maya se transparenta dejando ver la geometría formada en su interior.

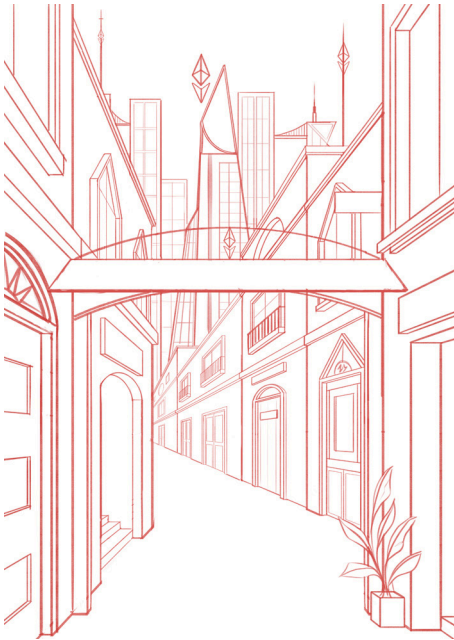


Figura 25. Boceto de distribución del escenario.
Por Azahar Ferrándiz Almagro.

2.3.3. Escenario

Para la puesta en escena se decidió representar una parte del casco antiguo de la ciudad. Una calle que discurre en diagonal con casas de construcción más antigua y que al fondo aparecen los edificios más modernos de *Aether*, entre ellos el edificio principal, con el símbolo del elemento flotando sobre él.

2.3.3.1. Modelado. A la hora de confeccionar este escenario se ha seguido un esquema similar al utilizado en Hollywood para crear decorados. Es decir, se omite la construcción de cualquier elemento que no aparezca en el encuadre o no aporte nada a la escena.

Sabiendo esto, se han ido modelando por separado cada uno de los elementos que componen la escena, separándolos en varias ventanas, puertas, los bloques que conforman los edificios y otros elementos visibles.

Esta etapa de modelado se ha caracterizado por el uso de superficies planas y aristas marcadas, prescindiendo del modificador *subdivision surface* utilizado anteriormente para algunos elementos de los personajes. También se ha utilizado el comando *bevel* para crear algunos arcos y biseles, y el modificador *array* para dar lugar a la repetición uniforme de elementos como tejas.

Una vez modeladas todas las piezas, se han colocado en su posición correspondiente y se han hecho pruebas con la cámara de la escena para obtener el encuadre deseado mediante la relación de aspecto y la distancia focal.

2.3.3.2. Texturizado. El proceso de texturado de los edificios ha consistido en seleccionar todas las caras de cada uno de ellos por separado y, en la pestaña de UV, se ha seleccionado la opción *create smart UV project*, dejando un pequeño margen entre las caras para evitar solapamientos de la textura. De esta manera las caras de cada objeto seleccionado se reparten ocupando el espacio proporcional que le corresponde dentro de la imagen utilizada para la textura. Tras esto se exportan los elementos en formato FBX.

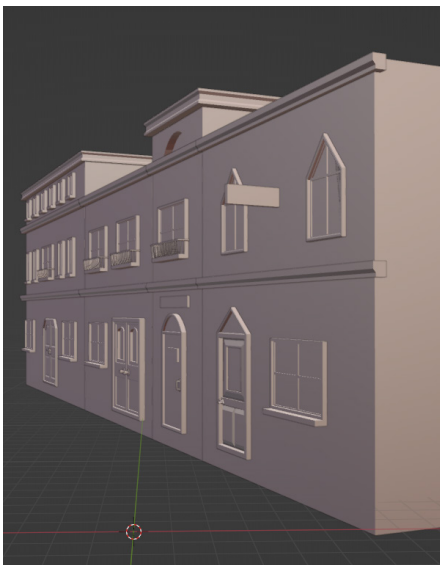


Figura 26. Modelado de los edificios de la sección intermedia de la escena.

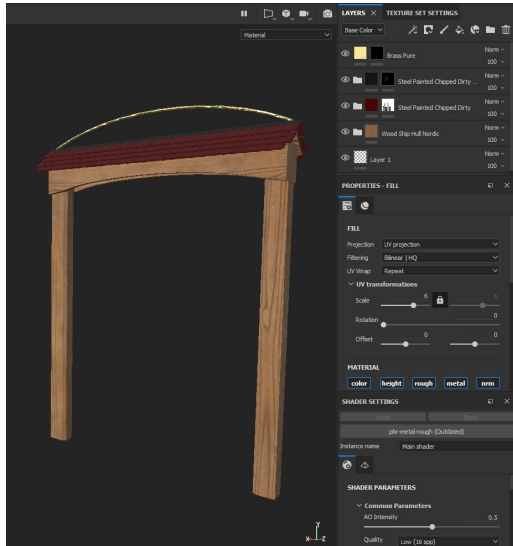


Figura 27. Proyecto de Substance Painter del arco principal de la escena.

A continuación, se han creado diversos proyectos de *Substance Painter* con una resolución de 2K dentro de los que se ha importado por separado cada uno de los objetos que hemos creado. Tras esto se comienza a aplicar las texturas correspondientes siguiendo un sistema de capas y máscaras similar al de Photoshop. Algunos de los materiales utilizados son los que ofrece por defecto *Substance Painter* y otros se han descargado de manera gratuita desde *Quixel*.

Por último, se exportan las texturas ya cocinadas a las carpetas correspondientes. Esto genera una imagen por cada uno de los siguientes parámetros: color base, mapa de desplazamiento, normales, rugosidad y metallic.

Se importan los modelos en *Blender* y se crean materiales para cada uno de los objetos. En el modo shading, se asignan las imágenes a los nodos correspondientes dentro de cada material.

2.3.3. Unreal Engine 5

Para finalizar con este proyecto, se ha montado el escenario y colocado a los personajes en *Unreal Engine 5*. Se tomó esta decisión porque se trata de un motor de videojuegos que recientemente ha implementado diversas tecnologías que representan avances considerables en la industria del 3D. Prueba de ello es la gran cantidad de videojuegos en desarrollo que utilizan este motor. Adicionalmente, hay producciones cinematográficas que en los últimos tiempos han aprovechado este programa para la grabación de películas en las que se proyectan escenarios envolventes.

Las principales características que han sido relevantes a la hora de realizar este proyecto han sido *Nanite* y *Lumen*. *Nanite* es una tecnología que permite realizar la retopología en tiempo real de los elementos estáticos mostrados en pantalla a partir de la resolución de la misma. De esta manera, se optimiza la carga gráfica a la que se ve sometido el hardware, ahorrando un paso que tradicionalmente se hacía de manera manual. Con el fin de probar esta función, los personajes no se han sometido a ningún tipo de retopología previa y, y que de manera conjunta, suman algo menos de un millón de polígonos.

Lumen es otra tecnología que utiliza el trazado de rayos para simular un comportamiento de la luz realista según cada material. Esto puede llegar a ser un inconveniente para ciertos ordenadores, ya que requieren una tarjeta gráfica de generaciones recientes para funcionar correctamente. Sin embargo, al tratarse de un proyecto en el que el procesado en tiempo real no es relevante, se ha podido aprovechar para iluminar la escena.



Figura 28. Estructura del escenario montado en Unreal Engine 5.

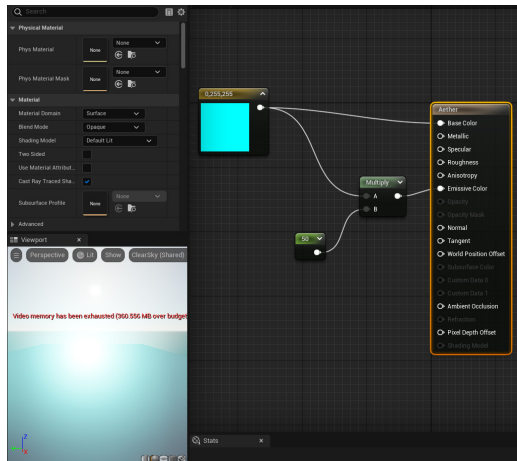


Figura 29. Sistema de nodos en Unreal Engine 5 para el material que emite luz azul.

2.3.4.1. Importación de recursos. La tarea de importar todos los recursos no ha presentado gran complicación, ya que desde Blender se han exportado los modelos 3D agrupados y con las texturas ya cocinadas y vinculadas a sus respectivos modelos.

Lo más importante la hora de importar cada elemento es que se seleccione la opción de utilizar *Nanite* para su geometría. De esta manera, los requisitos a la hora de ejecutar o renderizar serán menores.

Debido a que *Unreal Engine 5* utiliza un sistema de nodos diferente al que implementa *Blender*, ha habido errores con algunos materiales a la hora de importarlos directamente. Para solventar esto ha bastado con modificar los materiales importados y añadir las texturas necesarias con sus correspondientes nodos. Este problema se ha dado principalmente con los materiales que emiten luz, el cristal, y aquellos materiales que utilizan en sus nodos mapas de normales, metallic o roughness.

2.3.4.2. Luces. El objetivo para la puesta en escena era representar una escena de atardecer, en la que prima una luz general de color anaranjado, complementada con otras pequeñas fuentes de luz repartidas por el escenario de color azulado. Se optaron por estos colores ya que tradicionalmente es una paleta muy utilizada en cine para generar contraste, ya que ambos colores se complementan bien.

Para conseguir este efecto *Unreal Engine 5* implementa por defecto un sistema de iluminación realista basado en el comportamiento real de la iluminación natural. Entre otros parámetros, podemos modificar la posición del norte en nuestra escena, la ubicación geográfica mediante latitud y longitud, época del año especificando mes y día, y la hora solar, mediante la cual podemos posicionar el sol según el momento del día que se desee representar.

Para este proyecto se han modificado, principalmente, la ubicación del norte y la hora solar. De esta manera se ha conseguido ubicar la luz solar a la parte izquierda y dar un tono anaranjado a la escena.

Como elemento de contraste se han ido repartiendo los símbolos de Aether a que se les ha asignado un material azul emisor. De esta manera, se generan destellos y nuevas sombras que hacen que la composición cobre vida.

Otros elementos que aumentan la sensación de inmersión en la escena es el sistema de niebla exponencial, que difumina elementos lejanos, y el control de las nubes.



Figura 30. Parámetros utilizados para la iluminación global de la escena.



Figura 31. Ubicación de la cámara dentro de la escena de Unreal Engine 5.

2.3.4.3. Renderizado. Para la etapa final de este proyecto, ha sido necesario modificar los parámetros de la cámara para realizar el renderizado de la escena de manera correcta.

Se ha girado la cámara 90° para realizar el renderizado en formato vertical. Tras esto se ha asignado a la lente un objetivo simulado de 12mm con una apertura focal de 2.8. Esto ha sido necesario para que el render se asemeje lo máximo posible a la ilustración de referencia, abriendo considerablemente el ángulo del encuadre. Tras esto se ha modificado la relación de aspecto de la imagen, la cual por defecto se corresponde con la de la pantalla, es decir 16:9. Para este render se ha utilizado una configuración del carrete de 35mm Full Aperture.

Por último, ha sido necesario bajar la exposición de la cámara, ya que por defecto esta se adapta en tiempo real en *Unreal Engine 5*. Para evitar que el fondo apareciese quemado en el render final, se ha cambiado la exposición del render a modo manual, el cual se modifica en los parámetros de la cámara. Se ha definido una exposición de -1.0 para que todos los elementos se distinguen de manera correcta.

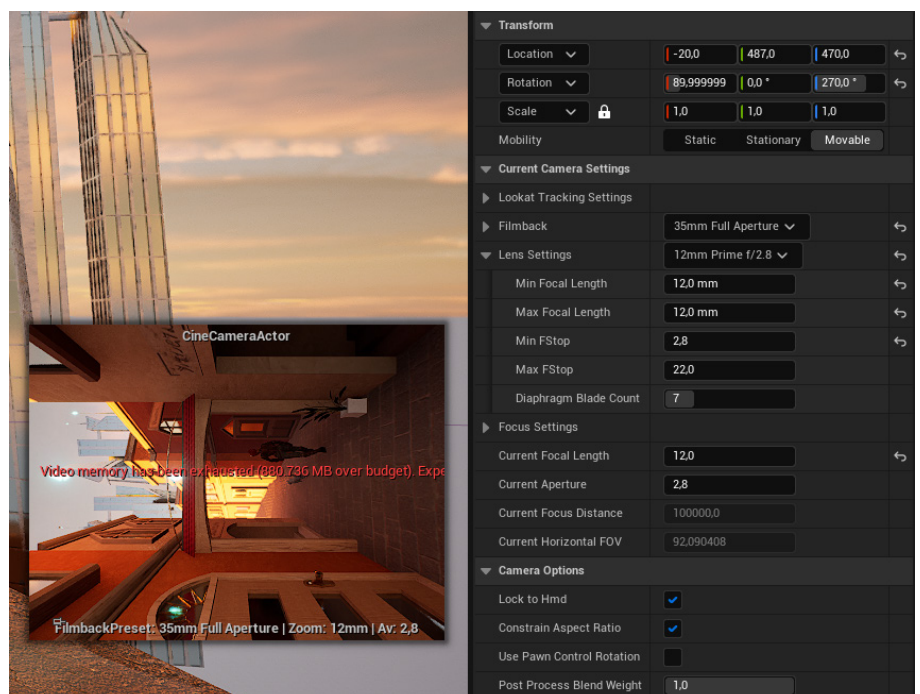


Figura 32. Parámetros utilizados en la cámara para el render final en Unreal Engine 5.

2.4. RESULTADO

Para presentar este proyecto se han realizado una serie de renderizados que muestran, por una parte, ambos personajes, y otro que muestra el conjunto de la escena con la perspectiva deseada.

Los primeros han sido realizados en *Blender*, utilizando *Cycles* como sistema de procesamiento de las luces para obtener un acabado más limpio. De esta manera, se resaltan más los detalles y se puede observar tanto a Galia como a Sake desde distintos ángulos.

El render final, que es el que refleja la ilustración que se deseaba obtener, ha sido realizado en *Unreal Engine 5*. En él se muestra el escenario, con los personajes ya situados y con las luces correspondientes para dar mayor sensación de inmersión y crear una imagen visualmente atractiva.

2.5. DIFUSIÓN

Teniendo en cuenta el resultado obtenido, se pretende adaptar este proyecto para su publicación en la *Workshop de Steam* en el futuro. En esta plataforma plataforma, los usuarios crean y difunden fondos de pantalla para que la comunidad los pueda utilizar en *Wallpaper Engine*, un programa que permite al usuario tener fondos de pantalla, animados o no, para sus ordenadores.

Para ello será necesario modificar el encuadre de la escena para mejorar la adaptabilidad con diferentes formatos de pantalla. También interesa implementar animaciones simples como partículas en suspensión o viento para seguir experimentando con el motor.

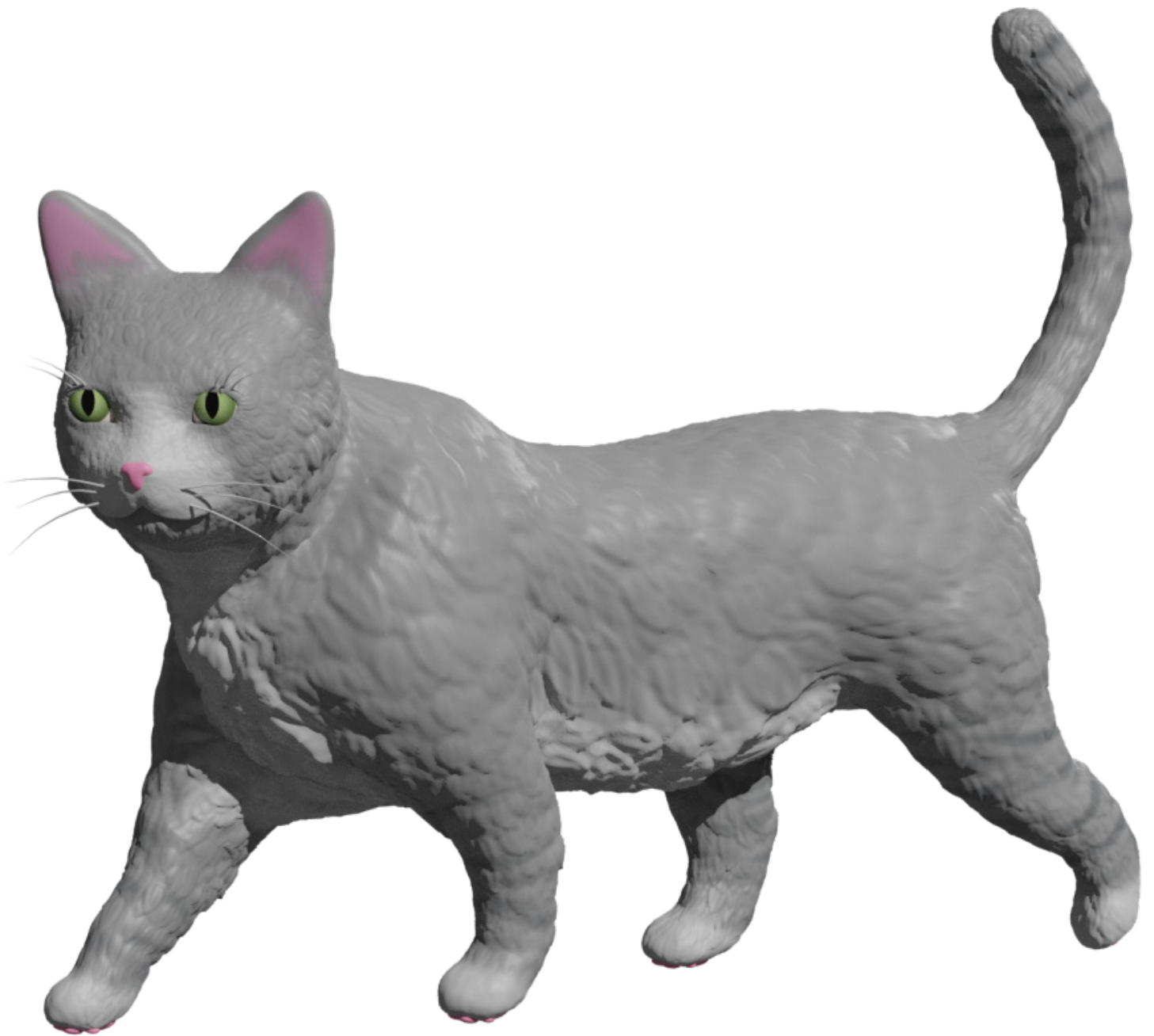
De esta manera, se pretende seguir desarrollando el proyecto implementando nuevas funciones y detalles y obtener feedback de usuarios reales.



Figura 33. Render de Galia y Sake.
Realizado en Blender utilizando Cycles.



Figura 34. Render de Galia.
Realizado en Blender utilizando Cycles.



*Figur 35. Render de Sake.
Realizado en Blender utilizando Cycles.*

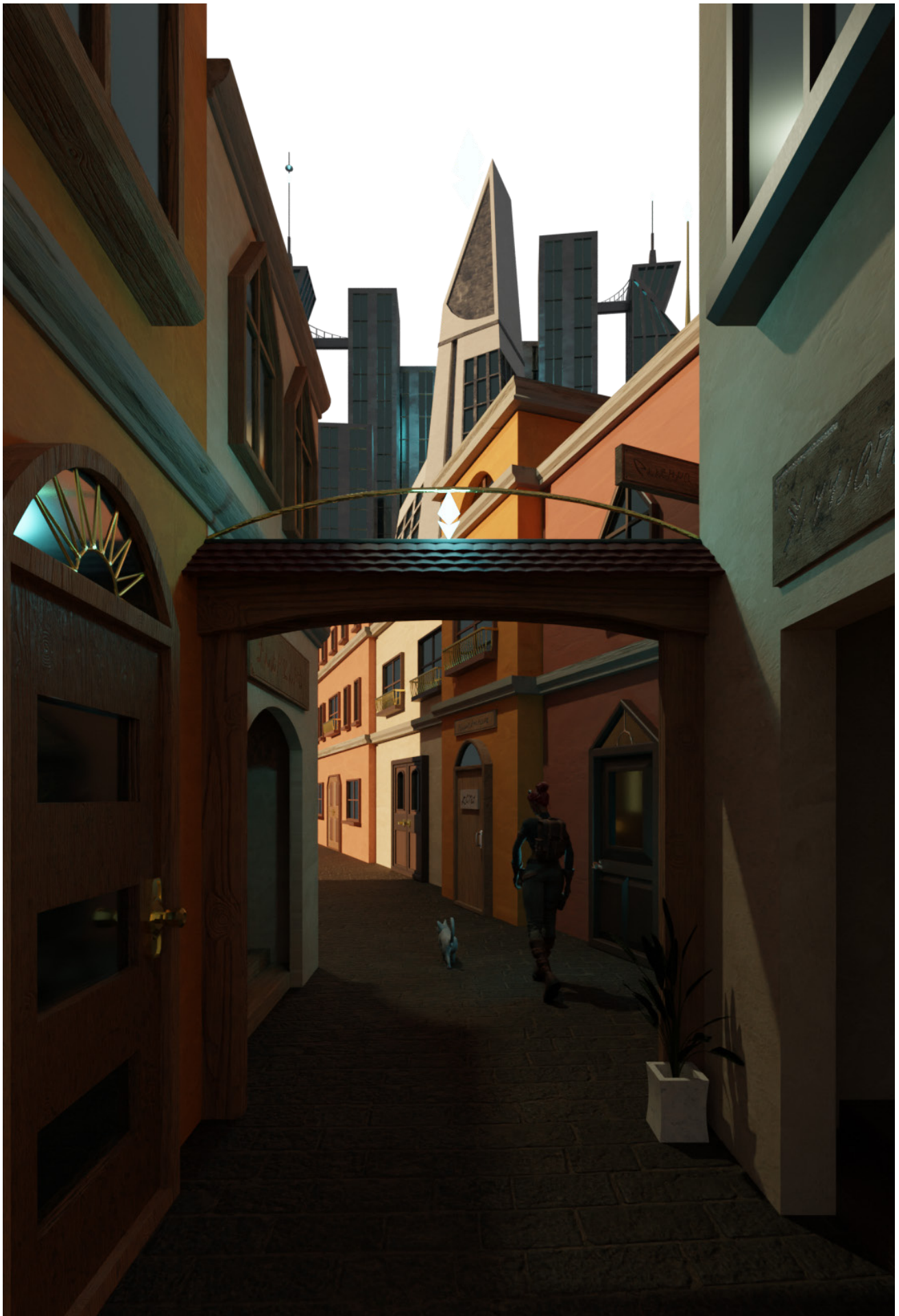


Figura 36. Render de prueba de la escena.
Realizado en Blender utilizando Cycles.



Figura 37. Render definitivo.
Realizado en Unreal Engine 5.

3. CONCLUSIONES

Tras haber finalizado este proyecto, ciertos aspectos dentro del flujo de trabajo han cobrado importancia sobre otros, así como la planificación previa y el uso adecuado de recursos. Por una parte, el haber sido consistente con la gestión de archivos y carpetas ha facilitado en gran medida el acceso a los mismos, haciendo el proceso de trabajo más cómodo y rápido.

Por otra parte, también ha sido esencial intentar ajustarse a los plazos propuestos para cada apartado y, en caso de no ser esto posible, adaptarse para recuperar el tiempo perdido. El trabajo coordinado ha sido, de igual manera, un factor determinante, ya que de esta manera este proceso se ha acercado al equivalente dentro del mundo profesional, donde artistas con diferentes especializaciones han de colaborar a la hora de realizar un proyecto.

Otro factor decisivo a tener en cuenta ha sido el apartado del hardware, que ha resultado ser limitante para los tiempos de renderizado y navegación por las interfaces de ciertos programas. Esto se debe a que el terminal utilizado para este proyecto es un ordenador portátil cuyos componentes no están todo lo preparados que deberían para este tipo de tareas. Con esto queda probado que disponer de las herramientas adecuadas facilita y acelera el trabajo a realizar.

Cabe destacar también la utilidad que ha tenido Unreal Engine 5 durante las fases finales del trabajo. Este motor ha demostrado hacer una muy buena gestión de recursos, justificando así su creciente uso dentro de la industria del cine, la animación y los videojuegos. Prueba de ello son los tiempos de renderizado, donde la misma escena ha tenido una media de tiempo de renderizado de 40 minutos en Blender frente a los 12 segundos que ha necesitado Unreal Engine 5 para renderizar una secuencia de 6 imágenes con resultados similares o incluso mejores.

El resultado ha sido el deseado y ha despertado el interés por seguir realizando proyectos similares. La falta de experiencia en algunos apartados se ha solventado con información online, que ha servido para adquirir y ampliar conocimientos, y se ha adquirido agilidad con el uso de los programas utilizados. Tanto los objetivos principales como los secundarios se han cumplidos y también se ha desarrollado en cierta medida la capacidad de trabajar en grupo para sacar adelante un proyecto.

4. BIBLIOGRAFÍA

WEBS GENERALES

Manual online de Blender 3.2: <https://docs.blender.org/manual/es/latest/>

Archivo de Unreal Engine 5: <https://docs.unrealengine.com/5.0/en-US/>

Página de Blender: <https://www.blender.org>

Página de Marvelous Designer: <https://www.marvelousdesigner.com>

Unreal Engine 5: <https://www.unrealengine.com/en-US>

Página de Riot Games: <https://www.riotgames.com/es>

Página de Valorant: <https://playvalorant.com/es-es/>

Página de Fortiche: <https://www.forticheprod.com>

Página de Arcane: <https://www.netflix.com/title/81435684>

Página de League of Legends: <https://www.leagueoflegends.com/es-es/>

RECURSOS

Página de Quixel: <https://quixel.com/megascans/home?category=surface&category=historical&search=floor>

Página de TurboSquid: https://www.turbosquid.com/?&utm_source=google&utm_medium=cpc&utm_campaign=ES-es-TS-Brand&utm_content=ts%20brand&utm_term=turbosquid&mt=e&dev=c&itemid=&targid=kwd-297496938642&loc=1005545&ntwk=g&dmod=&adp=<varies>&gclid=Cj0KCQjw2MWVBhCQARIsAljbwoMgxmgY6vxEk75Ui-QieHhFnTN-Vk8ZKN-Fdh2PCeIRjPktiMXAuQpwaAtvSEALw_wcB&gclsrc=aw.ds

Página de Mixamo: <https://www.mixamo.com/#/>

ARTSTATION

Página de Artstation de Minoh Kim: https://www.artstation.com/minoh_kim

Página de Artstation de Thibaut Granet: https://www.artstation.com/thibaut_granet

Página de Artstation de Yekaterina Bourykina: <https://www.artstation.com/ybourykina>

Página de Artstation de Simon Fuchs: <https://www.artstation.com/simonfuchs>

Página de Artstation de Brandon Martynowicz: <https://www.artstation.com/bmart>

YOUTUBE

- https://www.youtube.com/watch?v=x2t59qQzs_I&list=WL&index=11
- https://www.youtube.com/watch?v=wu_TkcQZqJg&list=WL&index=58&t=69s
- https://www.youtube.com/watch?v=0gLJlr_r3TE&list=WL&index=49
- <https://www.youtube.com/watch?v=yifhqPF4vu0&list=WL&index=53&t=704s>
- <https://www.youtube.com/watch?v=6I9VRGm45HE&list=WL&index=59&t=869s>
- <https://www.youtube.com/watch?v=bum0F00TaKU&list=WL&index=14>
- <https://www.youtube.com/watch?v=rkVOkFFYvIs>
- <https://www.youtube.com/watch?v=UY9nf6HEGIM&t=413s>

5. ÍNDICE DE FIGURAS

Figura 1. Concept final de Aether I. Por Azahar Ferrándiz Almagro.	10
Figura 2. Giro de Viper, de Valorant. Por Minoh Kim.	10
Figura 3. Giro de Jinx, de Arcane. Por Thibaut Granet.	11
Figura 4. Hoja de personaje de Caitlyn, de League of Legends. Por Yekaterina Bourykina.	11
Figura 5. Toronto, mapa de Overwatch. Por Simon Fuchs.	11
Figura 6. Split, mapa de Valorant. Por Brandon Martynowicz.	12
Figura 7. Esquema de organización de carpetas.	12
Figura 8. Diseño de Galia. Por Azahar Ferrándiz Almagro.	13
Figura 9. Malla base femenina. Por TheEmptyRoom.	13
Figura 10. Ajuste de proporciones de la malla según los bocetos.	13
Figura 11. Estructura base del pelo basada en puntos que indican la dirección.	13
Figura 12. Patronaje del vestuario junto a la orientación de las costuras.	14
Figura 13. Mallado definitivo de Galia, su vestuario y props.	14
Figura 14. Segmentación del rostro de Galia para trabajar los UVs.	14
Figura 15. UVs del cuerpo y rostro de Galia sobre su textura.	14
Figura 16. Texturas definitivas aplicadas a la malla de Galia.	14
Figura 17. Estructura de los huesos del rig de Galia. Generado con Mixamo.	15
Figura 18. Diseño de Sake. Por Azahar Ferrándiz Almagro.	15
Figura 19. Segunda versión de Sake con los primeros bocetos de fondo.	16
Figura 20. Mallado base definitivo de Sake.	16

Figura 21. Pelaje de Sake modelado con Alphas.	16
Figura 22. Segmentación de los UVs de Sake junto a su textura.	16
Figura 23. Texturas definitivas aplicadas a la malla de Sake.	17
Figura 24. Estructura de los huesos del rig de Sake. Generado con Rigify en Blender.	17
Figura 25. Boceto de distribución del escenario. Por Azahar Ferrándiz Almagro.	18
Figura 26. Modelado de los edificios de la sección intermedia de la escena.	18
Figura 27. Proyecto de Substance Painter del arco principal de la escena.	19
Figura 28. Estructura del escenario montado en Unreal Engine 5.	19
Figura 29. Sistema de nodos en Unreal Engine 5 para el material que emite luz azul.	20
Figura 30. Parámetros utilizados para la iluminación global de la escena.	20
Figura 31. Ubicación de la cámara dentro de la escena de Unreal Engine 5.	21
Figura 32. Parámetros utilizados en la cámara para el render final en Unreal Engine 5.	21
Figura 33. Render de Galia y Sake. Realizado en Blender utilizando Cycles.	23
Figura 34. Render de Galia. Realizado en Blender utilizando Cycles.	24
Figura 35. Render de Sake. Realizado en Blender utilizando Cycles.	25
Figura 36. Render de prueba de la escena. Realizado en Blender utilizando Cycles.	26
Figura 37. Render definitivo. Realizado en Unreal Engine 5.	27