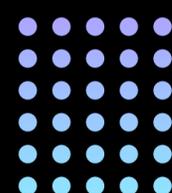


METAHUMAN UNLOCKED

Letner Hernández Gómez

Universidad LCI Veritas

Costa Rica, 2024



PRÓLOGO

En la última década, la industria de la animación y de los videojuegos ha avanzado notablemente, impulsada por innovaciones tecnológicas que han ampliado las fronteras creativas y técnicas. La creación de personajes de alta calidad se ha convertido en un elemento crucial para el éxito de cualquier producción interactiva, aunque sigue siendo un proceso complejo y costoso. El proyecto *MetaHuman Unlocked* surge para explorar soluciones que permitan democratizar estas herramientas, haciéndolas más accesibles y eficientes.

Este estudio se centra en la herramienta MetaHuman de Unreal Engine 5, con el fin de evaluar su capacidad para generar rigs de alta calidad en comparación a otros métodos tradicionales similares. A lo largo del proyecto, se fue desarrollando un personaje estilizado inspirado en la mitología japonesa, que pasó por un riguroso proceso de investigación, diseño conceptual, modelado 3D, texturizado, *rigging* e integración en Unreal Engine 5. La comparativa busca medir la calidad, eficiencia y adaptabilidad de los personajes creados con MetaHuman frente a los creados con una herramienta de *auto-rig* convencional, en este caso, Advanced Skeleton.

Los resultados de esta investigación proporcionarán una visión detallada de las capacidades y limitaciones de MetaHuman frente a Advanced Skeleton, además comparar los resultados obtenidos con cada herramienta en términos de tiempo, calidad, adaptabilidad y complejidad.

Este documento recoge exhaustivamente cada etapa del proyecto, desde su concepción hasta su evaluación final, con el objetivo de contribuir al avance del campo de la animación digital y ofrecer nuevas perspectivas y herramientas para la creación de personajes en videojuegos y otras producciones interactivas.

ÍNDICE

Capítulo 1: El diseño del personaje	04
· Anatomía del personaje	04
· Lineamientos del diseño	06
Capítulo 2: Deconstrucción de Metahuman	08
· El rig de Metahuman	08
· Las texturas de Metahuman	11
· La malla de Metahuman	13
Capítulo 3: Rigging en Advanced Skeleton 5	15
· Introducción	15
· Posicionamiento de los <i>joints</i>	17
· Pintado de pesos	19
· Rig facial	22
· Exportación e Importación a <i>Unreal Engine</i>	25
Capítulo 4: Rigging en Metahuman	29
· Introducción	29
· Proceso de <i>Wrapping</i> de la malla	30
· Proceso de <i>Baking</i> de texturas	33
· Pintado de pesos	37
· Exportación e Importación a <i>Unreal Engine</i>	39
Capítulo 5: Simulaciones y físicas en Unreal Engine	52
· Simulaciones de Pelo y Tela	52
· Configuración de físicas	57
Capítulo 6: Comparativa entre Metahuman y Advanced Skeleton	60
· Introducción	60
· Preservación estética	60
· Tiempos de desarrollo	61
· Adaptabilidad de la herramienta	61
· Usabilidad de la herramienta	61
· Calidad del rig	62
· Pruebas de animación	62
· Conclusiones	63

DISEÑO DEL PERSONAJE

1. Anatomía del personaje

El diseño de un personaje en la animación y los videojuegos es una tarea que requiere una comprensión profunda de la anatomía, independientemente de si este es altamente estilizado o busca un realismo extremo. Una constitución anatómica correcta no solo es fundamental para crear personajes creíbles y visualmente atractivos, sino que también juega un papel central en la funcionalidad y animación de estos.

Importancia de una anatomía correcta

Trabajar con una anatomía precisa permite que los personajes se muevan de manera natural y creíble. Esto es vital porque el público, aunque no siempre es consciente de los detalles técnicos, puede percibir fácilmente cuando algo no está bien en la estructura o los movimientos de un personaje. Una anatomía bien diseñada asegura que las articulaciones y los movimientos sean coherentes con la realidad, lo que mejora significativamente la inmersión y la conexión emocional del espectador con el personaje.

Además, un correcto diseño en este aspecto facilita el trabajo de animadores y *riggers*, ya que proporciona una base sólida sobre la cual se pueden aplicar diferentes técnicas de *rigging* y animación. La correcta distribución de la anatomía también asegura que el personaje pueda realizar una amplia gama de movimientos sin deformaciones no deseadas.

Consideraciones importantes

En el contexto de la creación de personajes, tanto MetaHuman de Unreal Engine 5 como herramientas de *auto-rigging* como Advanced Skeleton ofrecen una flexibilidad considerable. Esto quiere decir que no existen limitaciones estrictas que impidan trabajar con personajes de cualquier complejidad.

El personaje puede ser tan simple o detallado como se desee, sin embargo, es importante tener en cuenta que, a medida que aumenta la complejidad, también lo hace la necesidad de realizar ajustes específicos para lograr un resultado óptimo.

Para efectos de esta documentación, se ha decidido crear un personaje que desafíe significativamente los estándares típicos de MetaHuman, incorporando un grado considerable de estilización. Este enfoque tiene como objetivo demostrar que MetaHuman no es únicamente un creador de avatares hiperrealistas, sino que posee el potencial de integrar y gestionar personajes altamente estilizados y complejos. Pese a esta estilización, el personaje se mantendrá dentro de los lineamientos de ser bípedo y humanoide, para garantizar la compatibilidad y funcionalidad con las estructuras de MetaHuman.



Figura 1. Noboru. Arte conceptual de Noboru. Elaboración propia (2024).



Figura 2. Proporciones de Noboru. Hoja de diseño de Noboru. Elaboración propia (2024).

2. Lineamientos del diseño

Para maximizar la compatibilidad y el funcionamiento con MetaHuman de Unreal Engine 5, es preferible trabajar con personajes que sean bípedos, humanoides o antropomórficos. Estos tipos de personajes se ajustan mejor a las estructuras y *rigs* preestablecidos de MetaHuman, lo que facilita su proceso de integración y animación.

Aunque es posible adaptar personajes con características adicionales, como antenas, colas o extremidades extra, estos casos requieren modificaciones y ajustes específicos que escapan del alcance de esta documentación. Por consiguiente, el enfoque principal de este proyecto se centrará en personajes bípedos estándar, debido a que se procura que las técnicas y métodos descritos sean aplicables de manera eficiente y efectiva dentro de este marco.

Desafíos y metodología del proyecto

El personaje diseñado para este proyecto incluirá características avanzadas, como simulaciones de tela y de pelo, así como físicas aplicadas a algunas de sus prendas de vestir. Estas inclusiones buscan reflejar un *pipeline* de trabajo que abarque la mayoría de los escenarios posibles que podrían presentarse en una producción de videojuegos. A través de esta perspectiva, se pretende demostrar la capacidad de MetaHuman para manejar no solo avatares básicos, sino también personajes que requieren técnicas avanzadas y detalladas de animación y *rigging*.

Consideraciones de topología 3D

La topología 3D del personaje debe estar cuidadosamente diseñada para soportar una buena animación y *rigging*. Esto implica una distribución eficiente de polígonos, especialmente alrededor de las articulaciones y áreas de alta deformación, como los hombros, codos y rodillas. Una topología limpia y bien organizada facilita la tarea de aplicar *rigs* y animaciones complejas, evitando problemas como el estiramiento o la compresión inadecuada de la malla.

En este contexto, la topología debe permitir una flexibilidad que soporte tanto poses realistas como estilizadas. La densidad de la malla debe incrementarse en áreas que requieren mayor detalle y movimiento, mientras que se puede reducir en otras más rígidas o de menor visibilidad. Este equilibrio es clave para mantener la eficiencia del modelo sin comprometer la calidad visual y funcional del personaje.

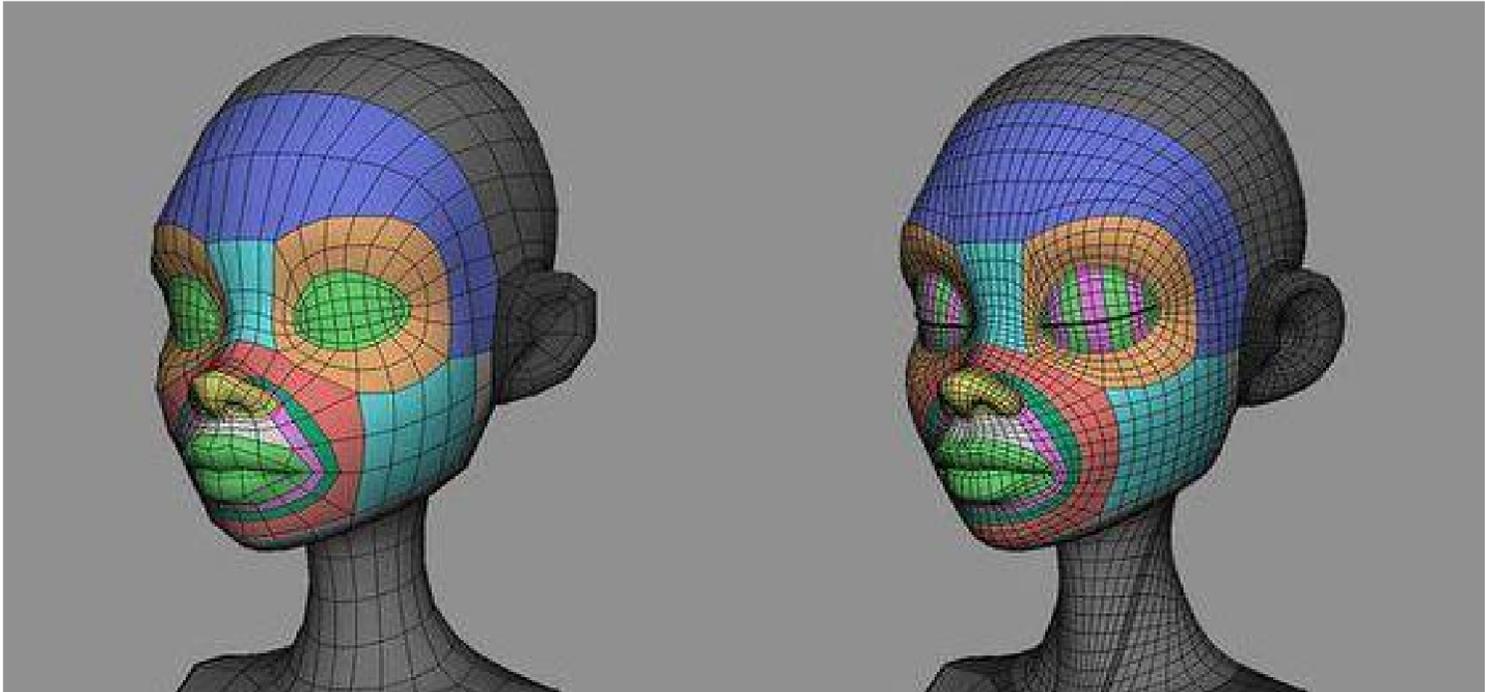


Figura 3. Facial Topology. Facial Modeling Timelapse. Recuperado de Vimeo: <https://vimeo.com/64302836> (2019).

Cuando se trata de MetaHuman, las consideraciones varían ligeramente; este aspecto será desarrollado con mayor profundidad más adelante. Sin embargo, es importante destacar desde este punto que el modelo 3D original tendrá que ser sometido a un proceso llamado *Wrapping*. Parte primordial del proceso de creación de un *metahuman* personalizado se centra en respetar su topología original. Por ende, aunque esta puede ser modificada en transformaciones básicas como rotación, escalado o traslación, no es posible aplicar modificaciones que alteren la cantidad de vértices sobre la malla. Todo esto también será abordado con detalle en el siguiente capítulo.

1. El *rig* de *Metahuman*

Para comprender a fondo el *rig* de *MetaHuman* y su potencial en la creación de personajes estilizados, primero se inspeccionará el *Blueprint* de un *metahuman* genérico. Este *Blueprint* está compuesto por diversos *Skeletal Meshes* que cumplen funciones específicas dentro de la estructura del personaje.



Figura 4. *Metahuman Demo Project*. Captura de pantalla de personaje genérico de *Metahuman*. Captura de pantalla del proyecto demo “*Metahuman*” de Epic Games (2024).

Un dato curioso e importante que se encontrará es que el *rig* corporal y el facial son dos *Skeletal Meshes* completamente distintos, cada uno con su propia jerarquía de *joints*. Esto significa que el *rig* del cuerpo y el de la cara operan de manera independiente dentro del *Blueprint*, lo cual permite una mayor flexibilidad y detalle en las animaciones faciales y corporales.

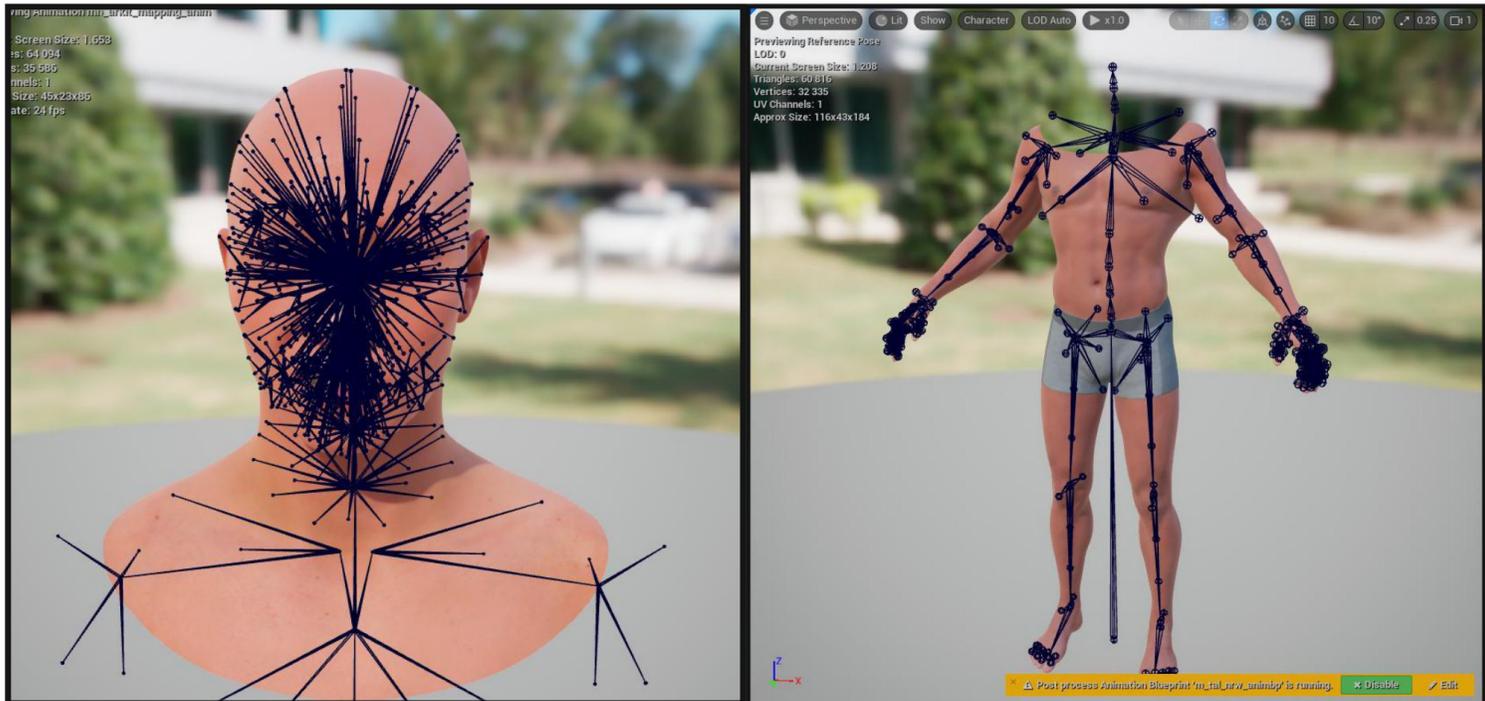


Figura 5. Metahuman Demo Project. Captura de pantalla de Skeletal Mesh de la demostración de Metahuman. Captura de pantalla del proyecto demo “Metahuman” de Epic Games (2024).

Además, se puede observar que, dentro del componente “*Body*” existen, varios componentes hijos designados para asignar las prendas de ropa del personaje. Estos facilitan la gestión y personalización del vestuario del *metahuman*, lo que permite intercambiar y modificar las prendas de manera eficiente.

Por otro lado, los componentes hijos de “*Face*” están conformados por *Groom assets*, que incluyen toda la cabellera y el vello facial del personaje. Estos *Groom assets* son cruciales para el realismo y la complejidad visual del personaje, ya que permiten una simulación avanzada de pelo y otros detalles finos que contribuyen a la apariencia general del *metahuman*.

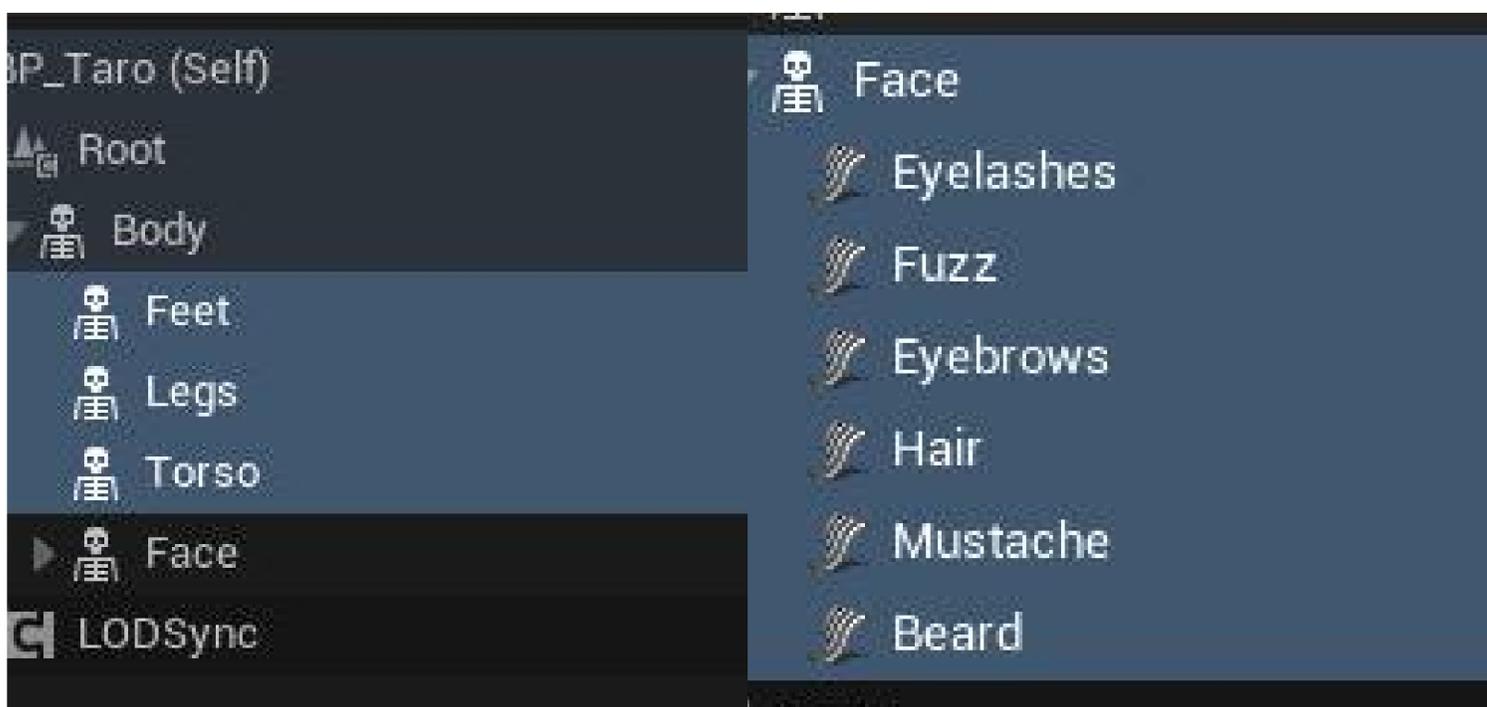


Figura 6. Metahuman Demo Project. Captura de pantalla de jerarquía del Blueprint de Metahuman. Captura de pantalla propia (2024).

Toda la estructura de *joints* corporales ya está configurada para reaccionar de forma realista a las deformaciones de los músculos anatómicamente correctos. Esta configuración permite que los movimientos del personaje sean naturales y convincentes, replicando las dinámicas musculares del cuerpo humano.



Figura 7. Metahuman Demo Project. Captura de pantalla de deformación muscular de Metahuman. Captura de pantalla del proyecto demo "Metahuman" de Epic Games (2024).

Asimismo, el *rig* facial cuenta con una gran cantidad de *joints* y *Morph targets* que controlan una inmensa variedad de expresiones faciales. Esta compleja red de controles faciales posibilita la creación de expresiones sutiles y realistas, vitales para transmitir emociones y mejorar la narrativa visual del personaje.



Figura 8. Metahuman Demo Project. Captura de pantalla de deformación facial de Metahuman. Captura de pantalla del proyecto demo "Metahuman" de Epic Games (2024).

2. Las texturas de *Metahuman*

En MetaHuman, las texturas juegan un papel clave en la creación de personajes realistas y expresivos. Una de las características distintivas de esta herramienta es su avanzado sistema de materiales, que está íntimamente ligado a la estructura del rig facial. Este sistema está compuesto por una serie de nodos que se interconectan con el rig facial para simular, de manera precisa, las arrugas y otros detalles que se forman en el rostro al realizar expresiones faciales específicas.

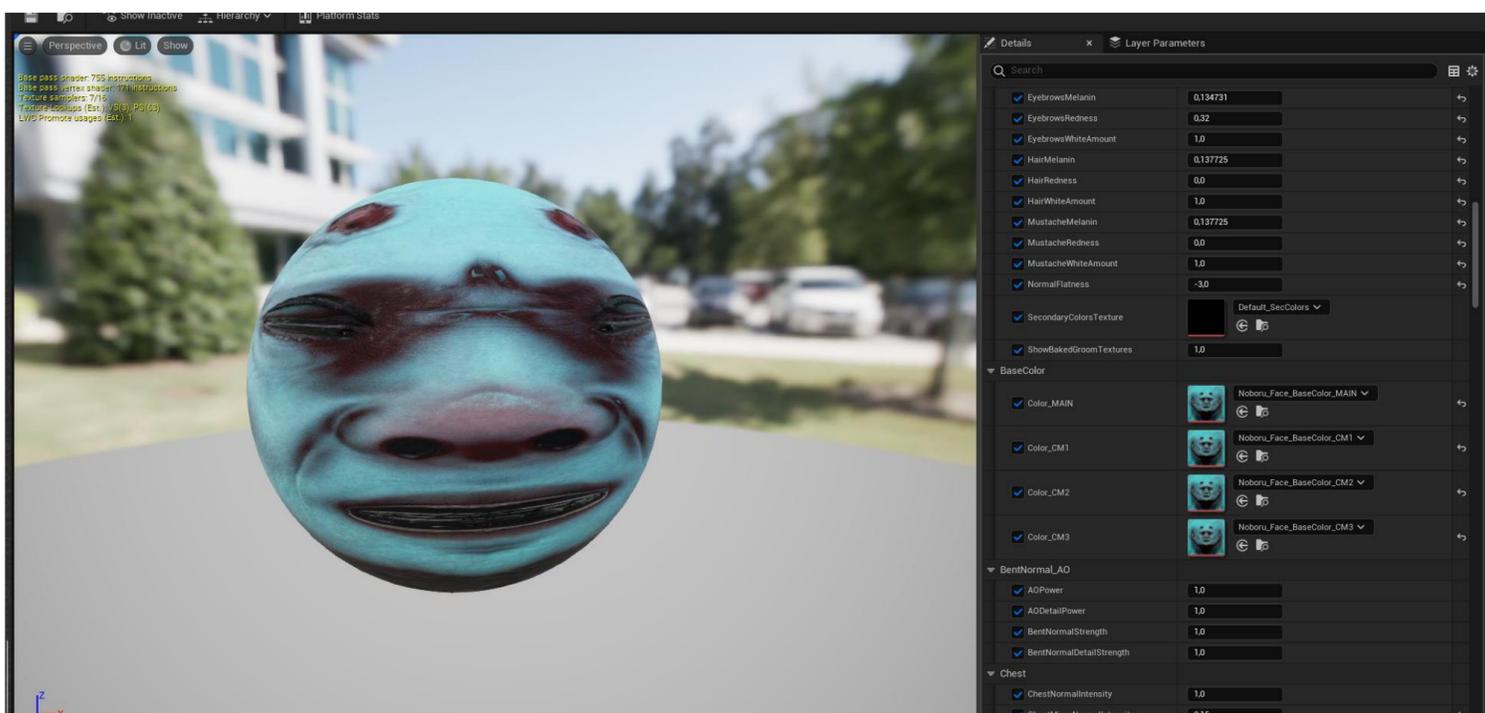


Figura 9. Noboru Material. Captura de pantalla de mapas de texturas sobre el material. Elaboración propia (2024).

El material del rostro de un *metahuman* está diseñado para estar en un constante intercambio de mapas de *Base Color* y normales. Estos se ajustan dinámicamente para resaltar o suavizar ciertas facciones del rostro dependiendo de la expresión que se esté realizando. Por ejemplo, al sonreír, el material puede activar un mapa de “Normal” que acentúa las líneas de expresión alrededor de la boca y los ojos, mientras que un ceño fruncido puede activar un diferente conjunto de mapas para marcar las arrugas en la frente.



Figura 10. Mapas de color y normales. Captura de pantalla de mapas de texturas de Metahuman. Captura de pantalla del proyecto demo "Metahuman" de EpicGames (2024).

Este intercambio dinámico de mapas es fundamental para mantener el realismo del personaje en movimiento. Los cambios sutiles en la textura del rostro permiten que las expresiones faciales se vean naturales y convincentes, evitando el aspecto estático que a menudo se observa en personajes menos avanzados.

A la hora de realizar las texturas para el *metahuman*, es indispensable tener en cuenta esta capacidad de intercambio de mapas. En este sentido, se deben preparar múltiples versiones de los mapas de *Base Color* y normales que se adaptarán a las distintas expresiones faciales del personaje. Estos mapas deben estar diseñados para integrarse sin problemas con el *rig* facial y ajustarse dinámicamente según las necesidades de la animación.



Figura 11. Comparativa de Wrinkles. Arrugas sobre la frente de un Metahuman. Captura de pantalla del proyecto demo "Metahuman" de EpicGames (2024).

3. La malla de *Metahuman*

Una de las particularidades de MetaHuman es su división en dos mallas separadas, al igual que sus dos *Skeletal Meshes* y *rigs* independientes. Este diseño específico es esencial para la funcionalidad avanzada de la herramienta y requiere consideraciones especiales al adaptar la malla de un personaje estilizado.

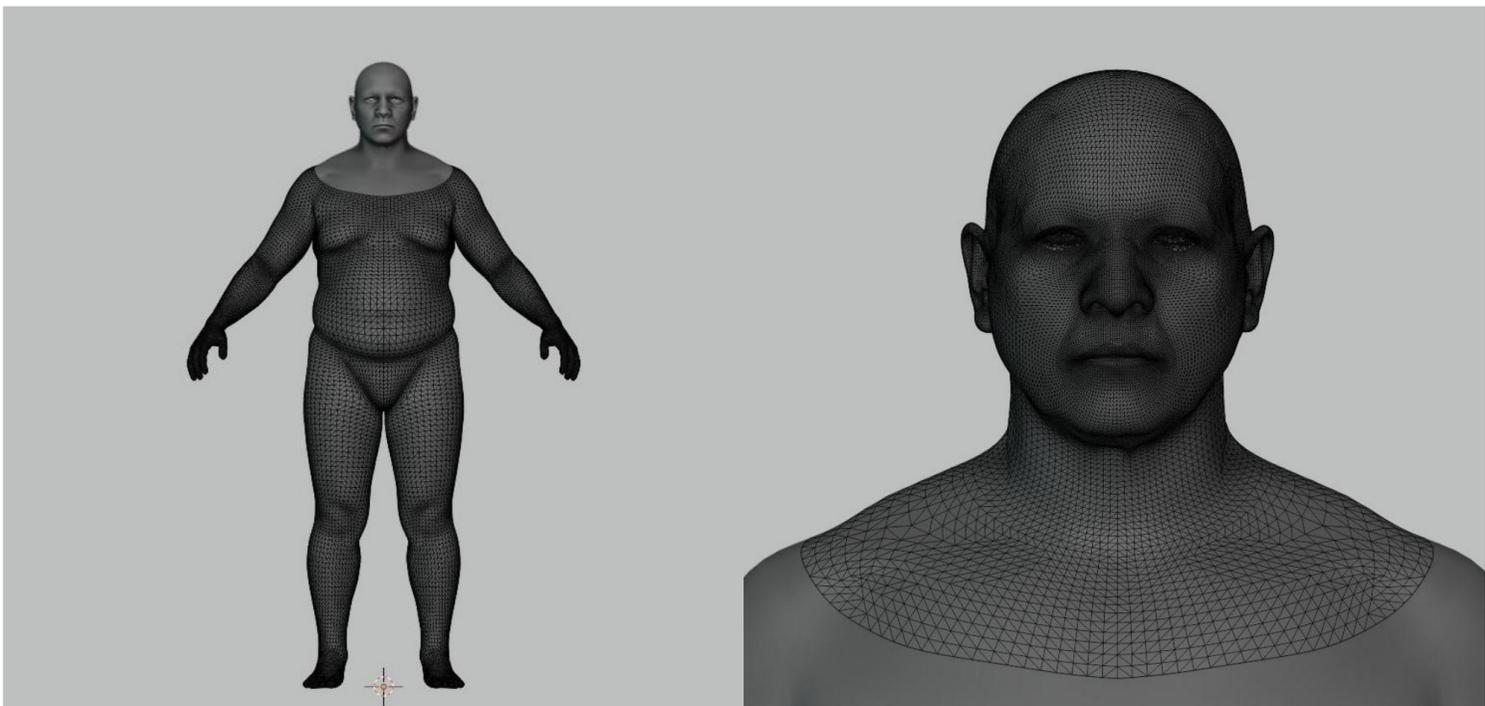


Figura 12. Metahuman en Blender. Captura de pantalla de inspección del modelo en Blender. Elaboración propia (2024).

Para modificar y adaptar la malla de este personaje estilizado a la estructura de *Metahuman*, es fundamental respetar el índice original de cada vértice. Cualquier modificación que altere este índice, como cortes de bucle (*loop cuts*), extrusiones, biselados (*bevels*) u otros modificadores que cambien la cantidad de vértices, resultará en un error de compatibilidad. Esta restricción es vital para mantener la integridad del rig y asegurar que las animaciones y deformaciones funcionen correctamente.

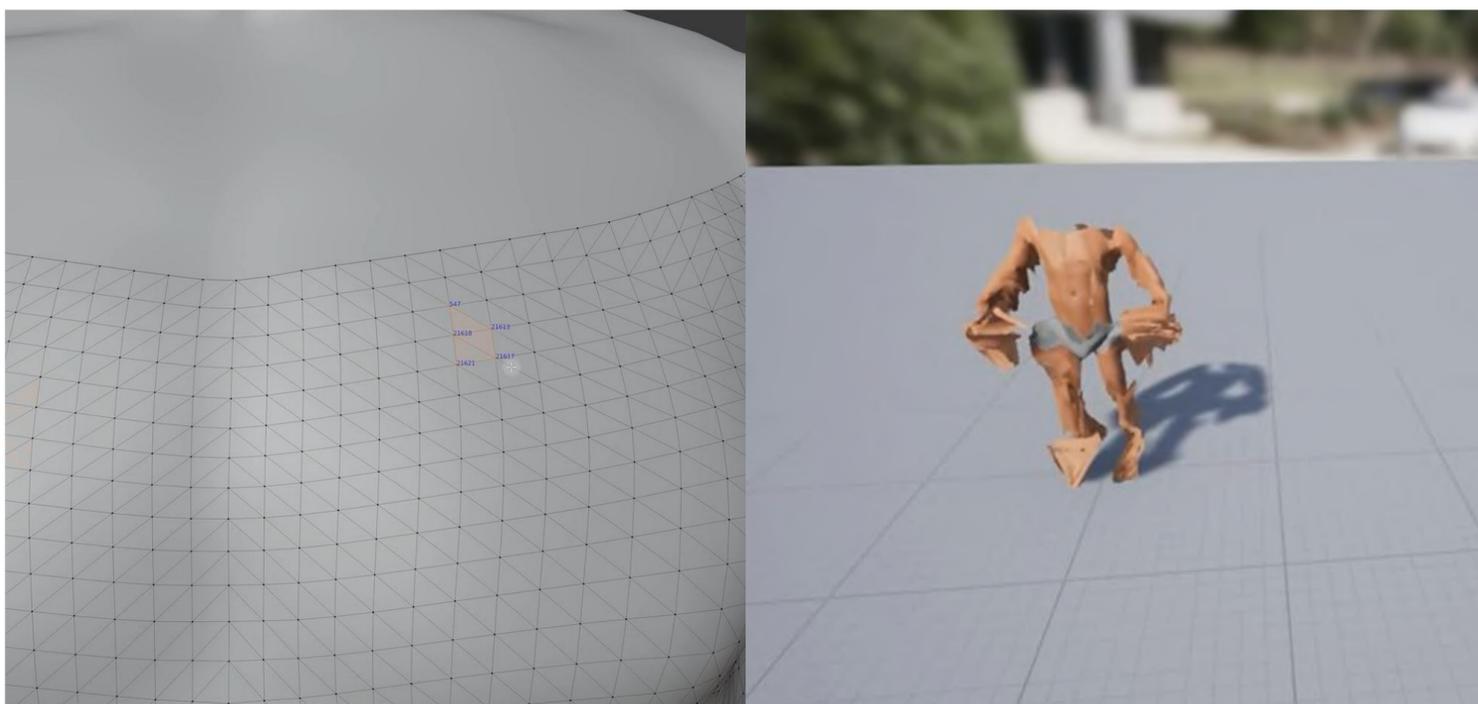


Figura 13. Metahuman en Blender. Captura de pantalla de inspección del modelo en Blender y Unreal Engine. Elaboración propia (2024).

En lugar de modificar directamente la malla a través de técnicas que cambian su topología, se deben emplear métodos que preserven la estructura original. Las herramientas recomendadas incluyen:

- *Blend Shapes / Shape Keys / Morph Targets*
- Herramientas de esculpido
- Traslación, rotación y escalado
- *Wrapping* de la malla personalizada sobre la malla de *MetaHuman*

Al utilizar estas técnicas, se garantiza que la malla del personaje estilizado se integre perfectamente con la estructura de MetaHuman. Esta consideración es esencial no solo para mantener la funcionalidad del rig, sino también para garantizar que todas las expresiones faciales y deformaciones corporales se reproduzcan de manera realista y precisa.



Figura 14. Metahuman en Blender. Captura de pantalla de Shape Keys del modelo en Blender. Elaboración propia (2024).

RIGGING EN *ADVANCED SKELETON*

1. Introducción

En cualquier producción de animación, la fase de *rigging* sigue a los procesos de modelado y texturizado. Este capítulo no se enfocará en cómo realizar estos últimos procedimientos de manera idónea, debido a que se asume que ya se han completado y que el modelo está listo para *rigging*. En cambio, se centrará en las capacidades de la herramienta convencional de *auto-rig*, *Advanced Skeleton*, la cual, aunque no es un rig desde cero con las herramientas nativas de Autodesk Maya, permite establecer un punto comparativo justo con respecto a *MetaHuman*.

Advanced Skeleton es una herramienta de *auto-rigging* ampliamente utilizada en la industria de la animación. Proporciona una solución rápida y eficiente para crear rigs complejos que pueden ser utilizados en diversas aplicaciones, desde videojuegos hasta cine. Su capacidad para generar rigs avanzados de manera semiautomática la convierte en una opción popular para estudios que buscan optimizar su pipeline de producción.

Para comenzar a trabajar con *Advanced Skeleton*, primero se debe descargar la última versión del plugin desde el sitio oficial *Advanced Skeleton – Download*. Una vez descargado, simplemente se arrastra el archivo titulado “*install.mel*” al *viewport* de Autodesk Maya para proceder con la instalación.



Figura 15. Instalación de *Advanced Skeleton*. Proceso de instalación. Elaboración propia (2024).

En este proyecto de Maya, se creará una escena que contenga al personaje con sus respectivas texturas y se agrupará toda la geometría en una carpeta denominada “geo”. En el menú de Advanced Skeleton, se podrá utilizar las opciones “*Model Clean*” y “*Model Check*”, ubicadas en la pestaña “*Preparation*”. Estas opciones son esenciales para preparar el modelo para el *rigging*.

Generalmente, se busca que el modelo sea completamente simétrico para evitar cualquier tipo de errores. Sin embargo, en caso de que no lo sea, es crucial que el eje de simetría esté perfectamente alineado al centro. En este proyecto, se pudo ignorar todos los errores, y lo único que fue necesario fue alinear perfectamente el eje central del modelo.

Además, se debe garantizar que el punto de pivote de este personaje esté alineado en el eje (0,0,0) del mundo. De lo contrario, Advanced Skeleton enviará una advertencia. Por su parte, se recomienda que la geometría de los ojos, cejas y pestañas esté separada y que exista una versión derecha y otra izquierda para facilitar el *rigging*.

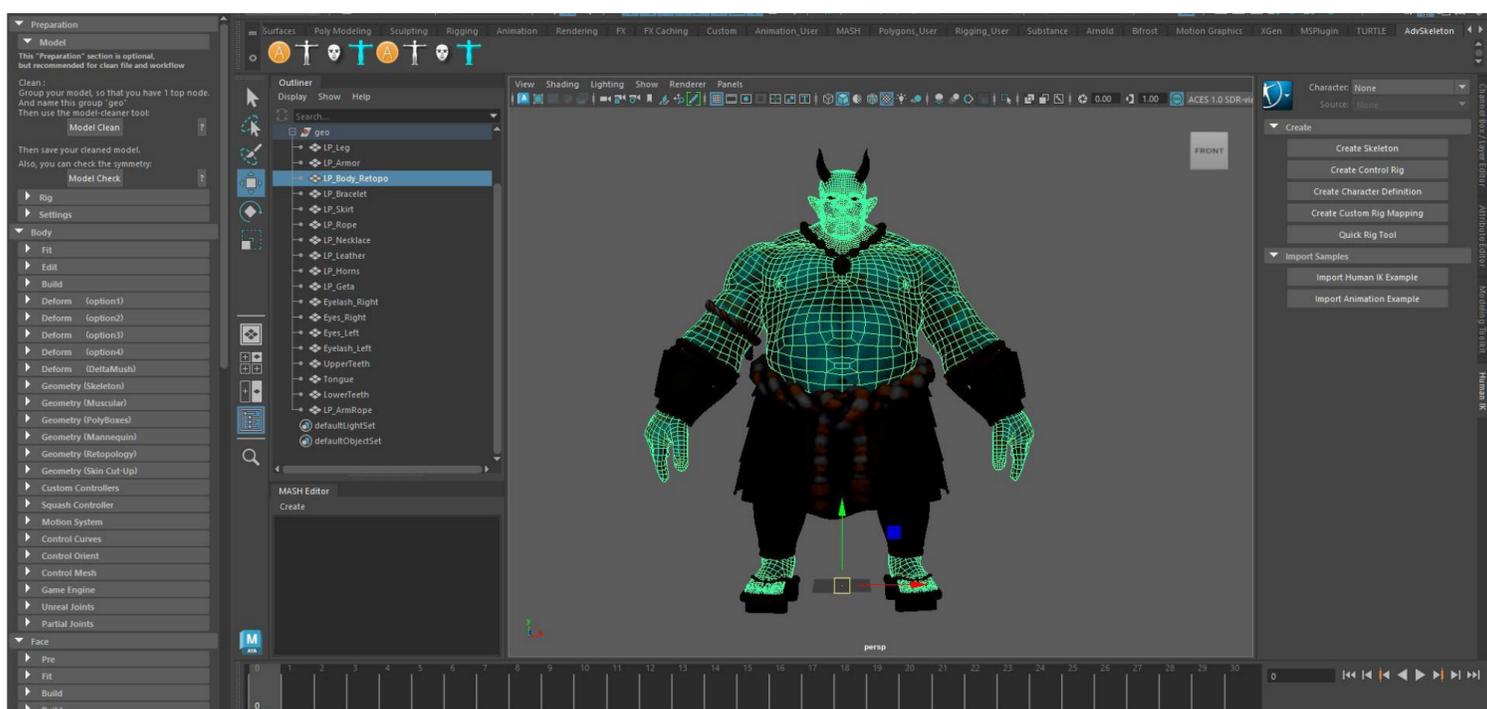


Figura 16. Interfaz de Advanced Skeleton. Proceso de rigging de Noboru. Elaboración propia (2024).

Una vez aplicadas estas consideraciones, se puede guardar esta escena y crear una nueva, completamente limpia, dentro de este proyecto. Este paso es fundamental para mantener un entorno de trabajo organizado y evitar posibles conflictos o errores durante el proceso de *rigging*.

2. Posicionamiento de los *Joints*

El primer paso en el proceso de *rigging* con Advanced Skeleton es crear una primera escena como referencia. Este método permite tener el modelo original vinculado a la escena de *rigging*, de manera que cualquier cambio necesario en el modelo puede realizarse directamente en la escena de referencia y se reflejará automáticamente en la de *rigging*. Esto garantiza un flujo de trabajo eficiente y dinámico.

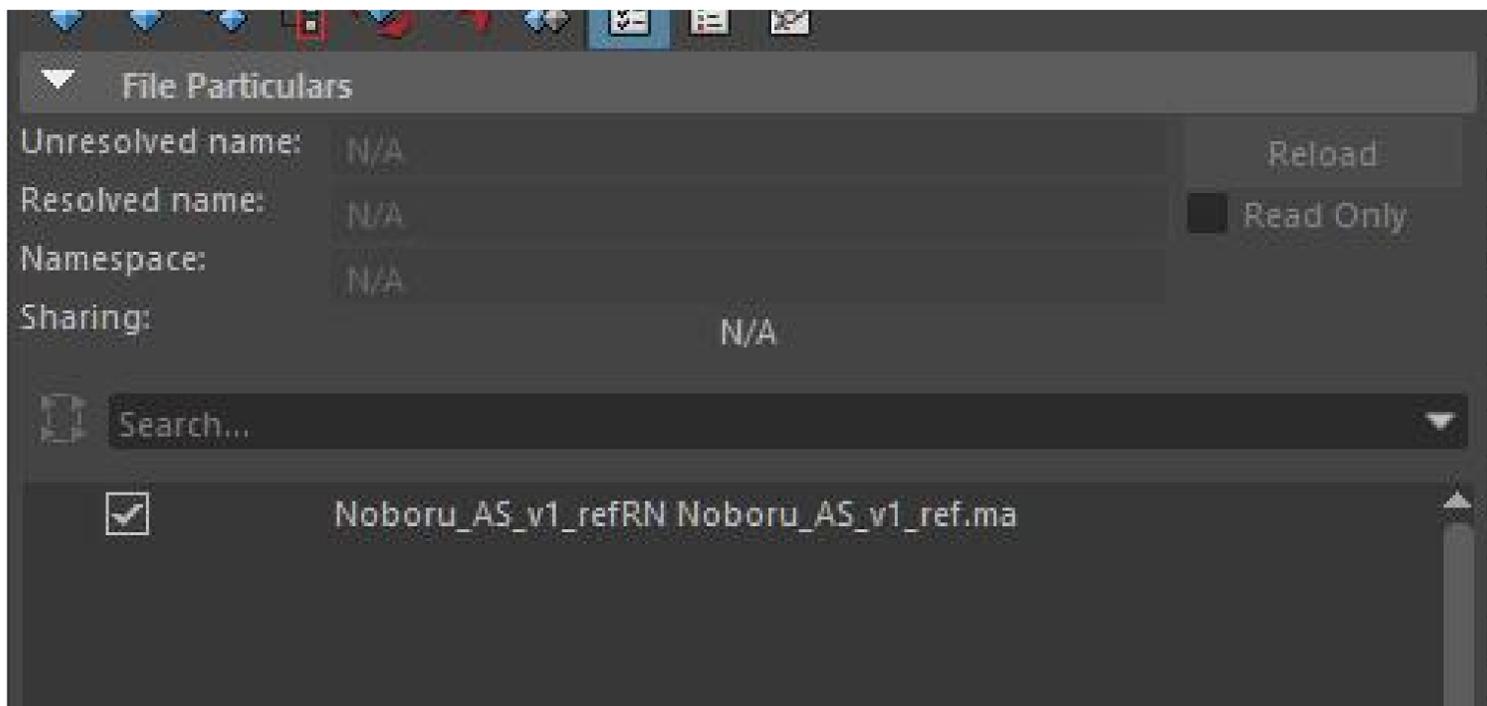


Figura 17. Interfaz de Advanced Skeleton. Referencias del proyecto. Elaboración propia (2024).

Una vez creada la referencia, se procederá al apartado de "*Body-Fit*" y se seleccionará la opción para importar el archivo *biped.ma*. Esta acción importará la estructura inicial del *rig* corporal. La estructura de *joints* y controles necesarios para el *rigging* de un personaje bípedo se cargarán en la escena.

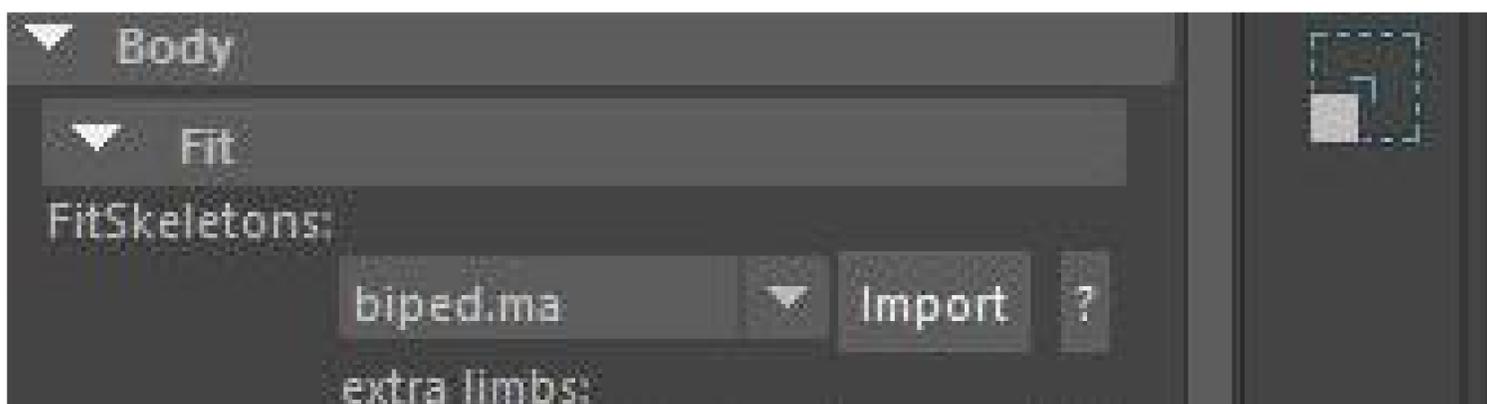


Figura 18. Interfaz de Advanced Skeleton. Importación de plantilla. Elaboración propia (2024).

El siguiente paso es escalar esta estructura para que coincida con el tamaño del modelo 3D. En este sentido, se utilizarán las herramientas de traslación y rotación para posicionar cada *joint* en su lugar correspondiente. Este punto es esencial y puede ser bastante detallado y laborioso; por lo tanto, es importante realizarlo con paciencia y precisión para corroborar que cada *joint* esté correctamente alineado con la anatomía del modelo.

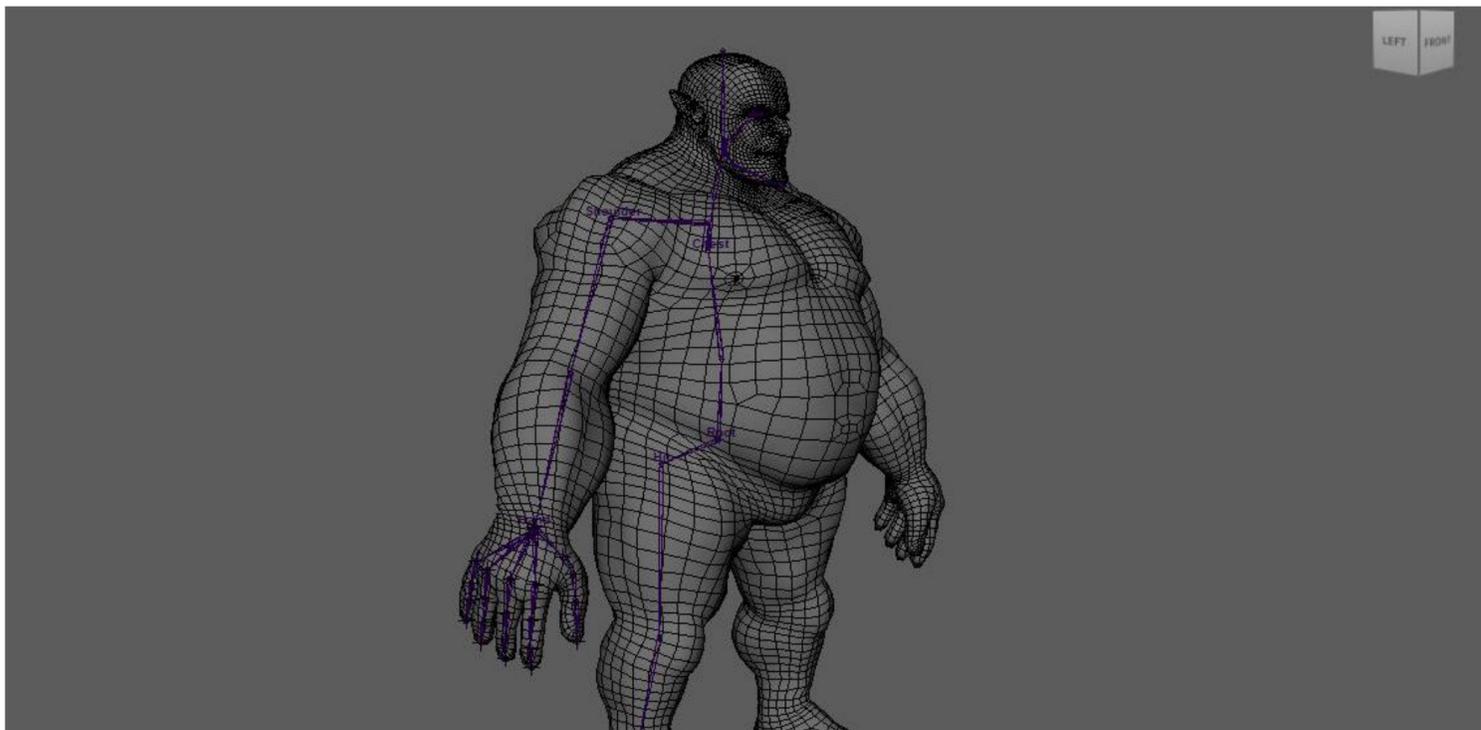


Figura 19. Interfaz de Advanced Skeleton. Proceso de rigging de Noboru. Elaboración propia (2024).

Una vez que se han posicionado correctamente todos los *joints*, se guardará una versión del archivo para mantener un respaldo, antes de proceder al siguiente paso. Seguidamente, es necesario dirigirse a la pestaña “*Build*” en el menú de *Advanced Skeleton* y presionar el botón “*Build Advanced Skeleton*”. Esta acción generará todos los controladores necesarios para el rig corporal.

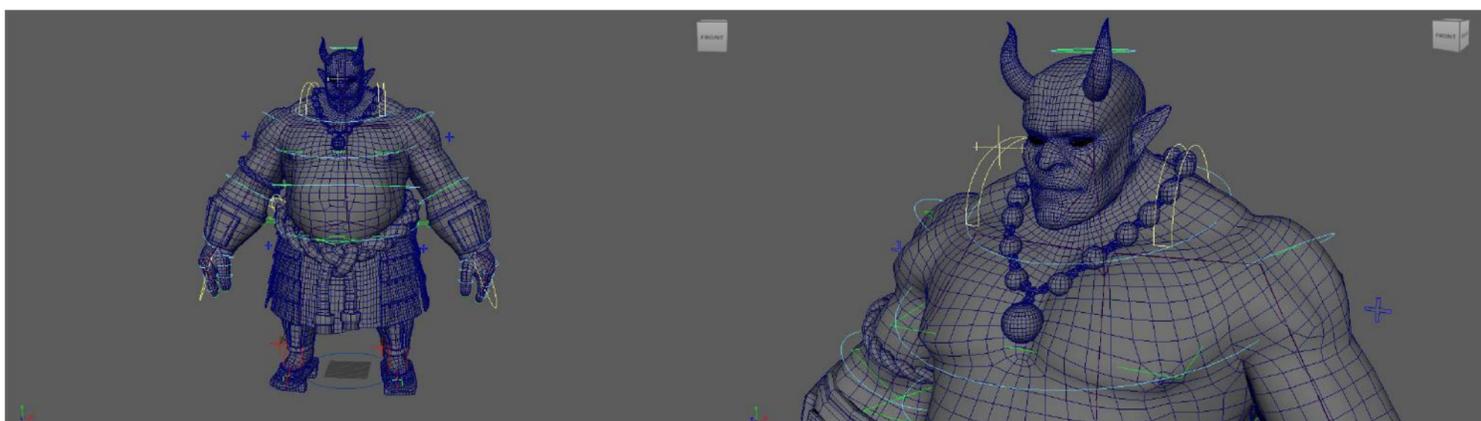


Figura 20. Interfaz de Advanced Skeleton. Proceso de rigging de Noboru. Elaboración propia (2024).

Es fundamental aprovechar este momento para revisar minuciosamente la orientación y funcionalidad del rig. Cualquier ajuste necesario en la orientación o en la posición de los *joints* debe realizarse antes de continuar con el proceso. Esta revisión detallada asegura que el rig funcionará correctamente y evitará problemas en etapas posteriores del proyecto.

3. Pintado de pesos

Cuando se hayan realizado todos los ajustes necesarios sobre el esqueleto y sus controladores estén colocados adecuadamente, el siguiente paso es el pintado de pesos, un proceso clave para asegurar que la deformación de la malla sea precisa y natural al moverse. Para comenzar, hay que dirigirse a la pestaña “*Deform Option 2*” en el menú de Advanced Skeleton.

En esta sección, primero se va a crear un *Skin Cage*. El *Skin Cage* es una estructura de bloques sencillos conectados al propio rig, que permitirá transferir los pesos a la malla principal del personaje de manera controlada y eficiente.

- **Crear el *Skin Cage*:** Se selecciona la opción para crear un *Skin Cage* desde el menú. Esto generará una estructura simplificada alrededor del personaje, que se conectará automáticamente con el *rig* que ya se ha configurado.



Figura 21. Interfaz de Advanced Skeleton. Proceso de pintado de pesos. Elaboración propia (2024).

- **Transferencia de pesos:** Después de crear el *Skin Cage*, se selecciona la malla del personaje y se presiona el botón “*Copy Weights*”. Esta acción transferirá los pesos de la malla del Skin Cage a la malla principal, aplicando una distribución inicial de pesos que refleja la influencia de cada *joint* en la geometría del personaje.

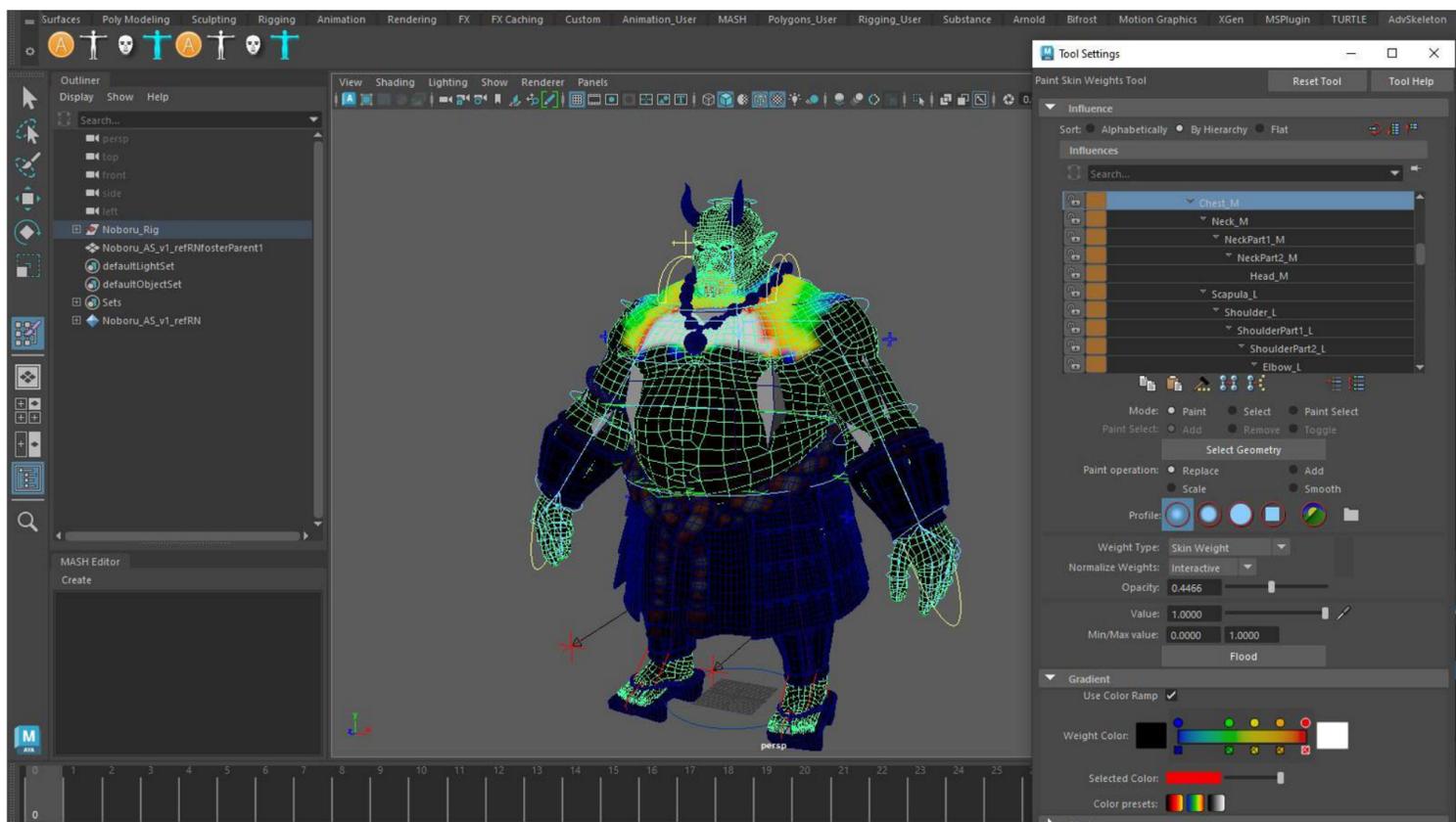


Figura 22. Interfaz de Advanced Skeleton. Proceso de pintado de pesos. Elaboración propia (2024).

Ajuste de pesos

Con los pesos iniciales aplicados, el rig corporal estará casi listo. Sin embargo, se notará que, al mover los controladores, la deformación de la malla puede no ser perfecta. Para mejorar y refinar estos pesos, se utilizarán varias herramientas disponibles en Autodesk Maya:

- **Paint Skin Weights Tool:** Esta herramienta permite pintar los pesos directamente sobre la malla, controlando la influencia de cada *joint* con un pincel virtual. Se puede agregar o quitar peso para asegurar que la deformación sea suave y precisa.
- **Hammer Tool:** Es útil para ajustar rápidamente los pesos en áreas específicas de la malla, lo cual proporciona una manera rápida de corregir pequeñas imperfecciones.
- **Smooth Weights Tool:** Se utiliza para suavizar las transiciones entre diferentes influencias de *joints*, garantizando que la deformación de la malla sea fluida y natural.

Nuevamente, es importante abordar este proceso con paciencia y atención al detalle. El pintado de pesos puede ser un proceso minucioso, pero es esencial para comprobar que el personaje se deforme correctamente durante la animación.

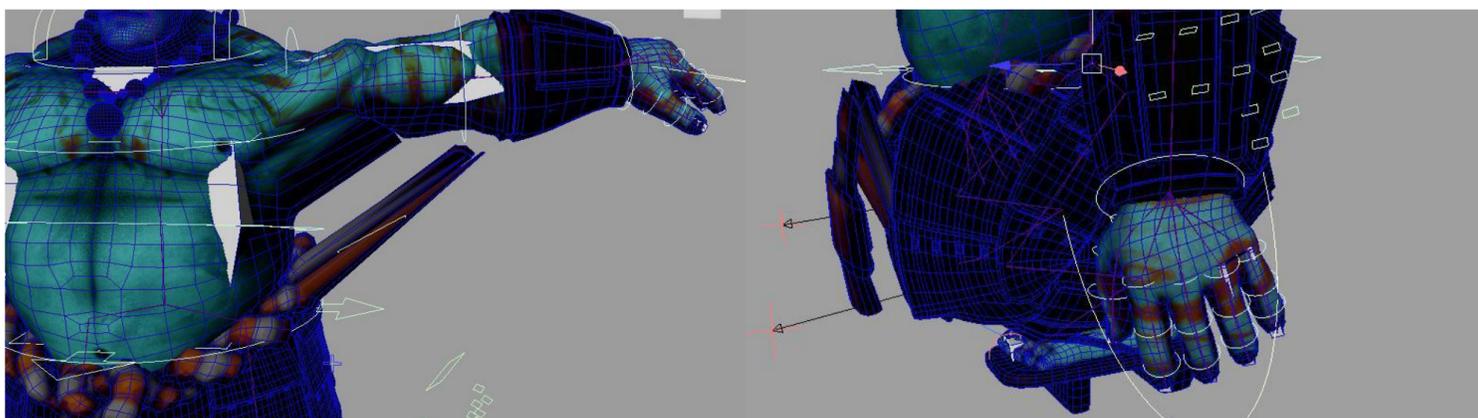


Figura 23. Interfaz de Advanced Skeleton. Proceso de pintado de pesos. Elaboración propia (2024).

Finalización del *rig* corporal

Cuando se hayan pintado completamente los pesos y se esté satisfecho con los resultados, se tendrá un *rig* corporal completamente funcional. Esto significa que el personaje responderá adecuadamente a los movimientos de los controladores y proporcionará una base sólida para cualquier animación futura.

Con el *rig* corporal finalizado, se estará listo para proceder a la siguiente etapa del *rigging*: el *rig* facial. Este paso permitirá añadir un nivel adicional de control y expresividad al personaje, de manera que el proceso de *rigging* quede completo y el personaje esté preparado para su integración en la producción.

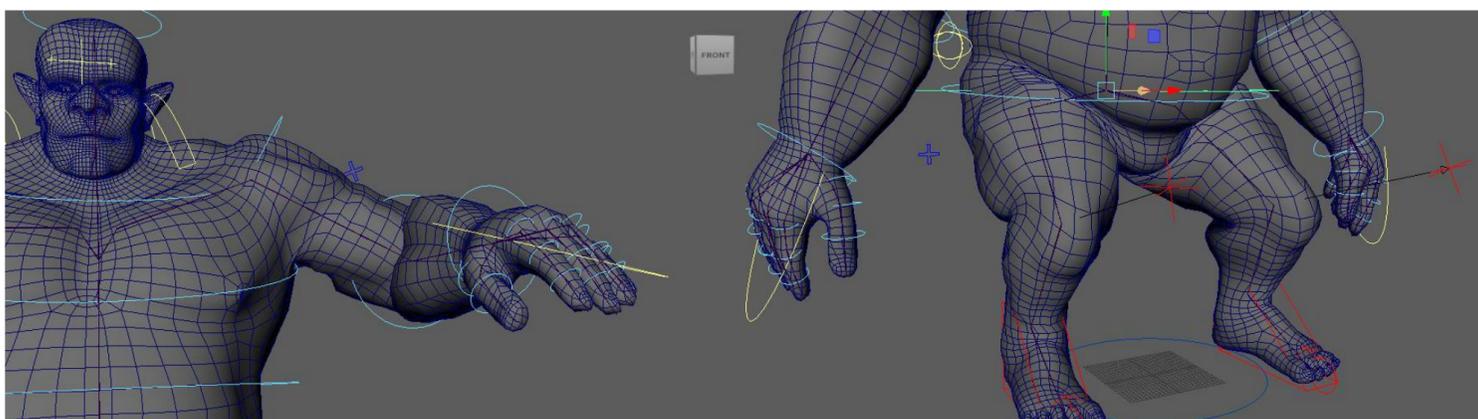


Figura 24. Interfaz de Advanced Skeleton. Deformaciones musculares de Noboru. Elaboración propia (2024).

4. Rig facial

El rig facial es uno de los apartados más sencillos de configurar gracias a las múltiples herramientas que *Advanced Skeleton* ofrece para cumplir esta función de forma eficiente. A continuación, se detallan los pasos para crear el rig facial.

Configuración Inicial

- **Configuración del apartado "Face - Pre":** Es necesario dirigirse al apartado "Face - Pre" en el menú de *Advanced Skeleton*. Aquí, se deberá completar todas las casillas requeridas. Se puede utilizar el botón con el signo de interrogación (?) para obtener ayuda adicional sobre la información necesaria para cada campo.

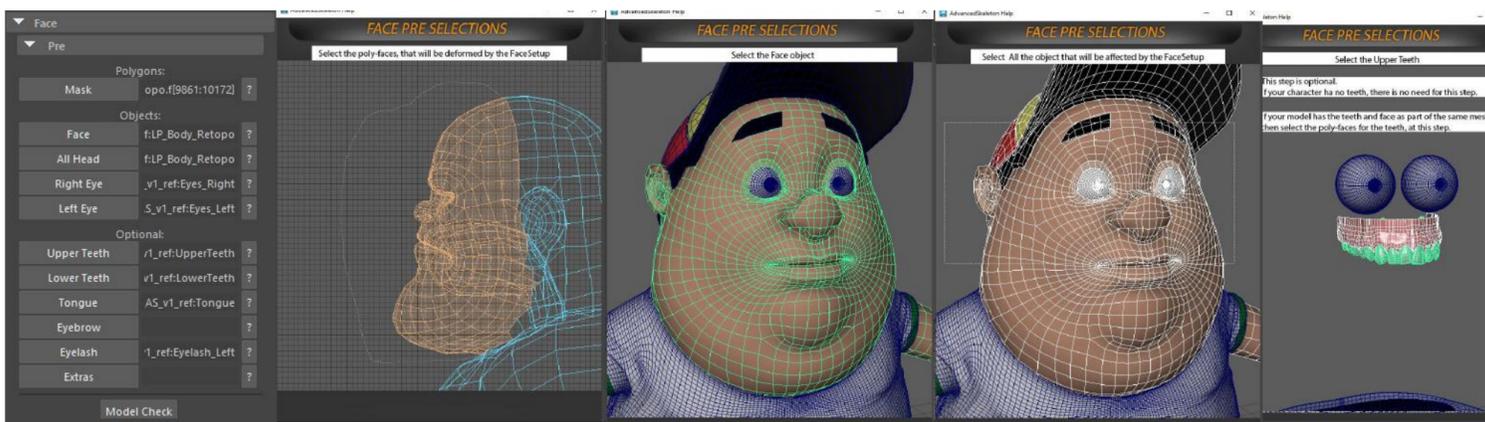


Figura 25. Interfaz de *Advanced Skeleton*. Sección Help. Captura de pantalla de interfaz de *Advanced Skeleton* (2024).

- **Selección del tipo de rig:** En el apartado de selección del tipo de rig, es crucial elegir la opción "*Blend Shapes*". En la actualidad, esta es la única forma de exportar el rig facial de *Advanced Skeleton* a *Unreal Engine 5* de modo que se garantice la compatibilidad y funcionalidad.



Figura 26. Interfaz de *Advanced Skeleton*. Captura de pantalla de selección del tipo de rig facial. Elaboración propia (2024).

Configuración del apartado "Fit"

- **Completar las casillas del apartado *Fit*:** Una vez configurado el apartado "Face – Pre", hay que dirigirse a "Fit" y completar todas las casillas necesarias. Si el modelo no es completamente simétrico, es preferible realizar el ajuste tanto para el lado izquierdo como para el derecho del rostro, para asegurar una correcta simetría y funcionalidad del rig facial.

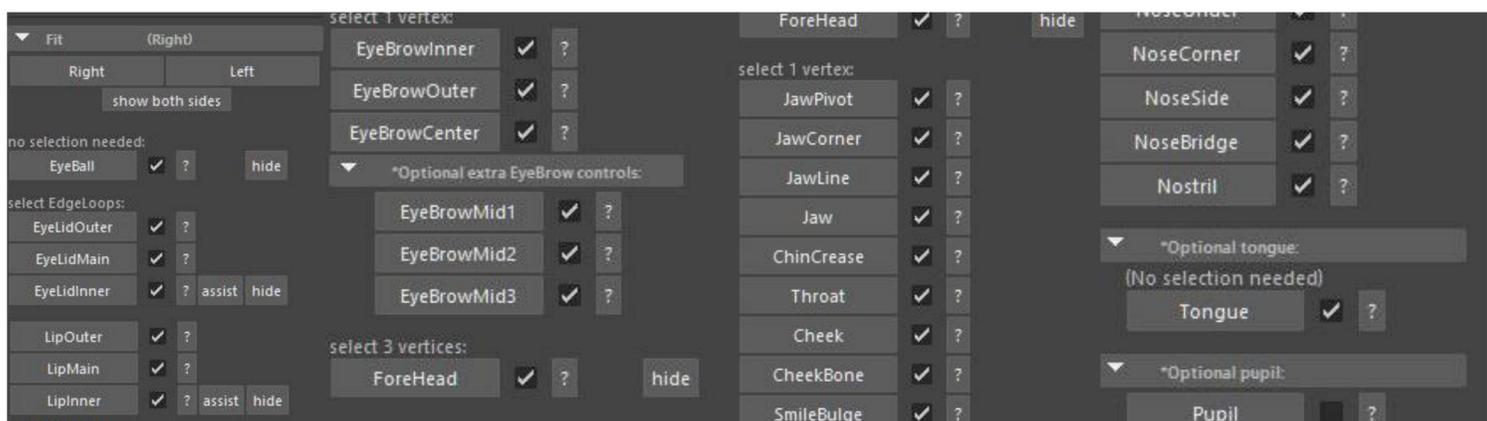


Figura 27. Interfaz de Advanced Skeleton. Captura de pantalla de la sección de ayuda del rig facial. Elaboración propia (2024).

Construcción del rig facial

- **Precauciones:** Una vez configuradas todas las opciones, debe guardarse una versión del archivo antes de proceder. Esto es importante, ya que, si ocurre algún error durante el proceso, será posible regresar a esta versión y corregir cualquier problema.
- **Build Advanced Face:** Presiona el botón "Build Advanced Face". Advanced Skeleton comenzará a crear el rig facial automáticamente. Este proceso generará los controladores necesarios y aplicará los ajustes pertinentes para un rig facial funcional y eficiente.

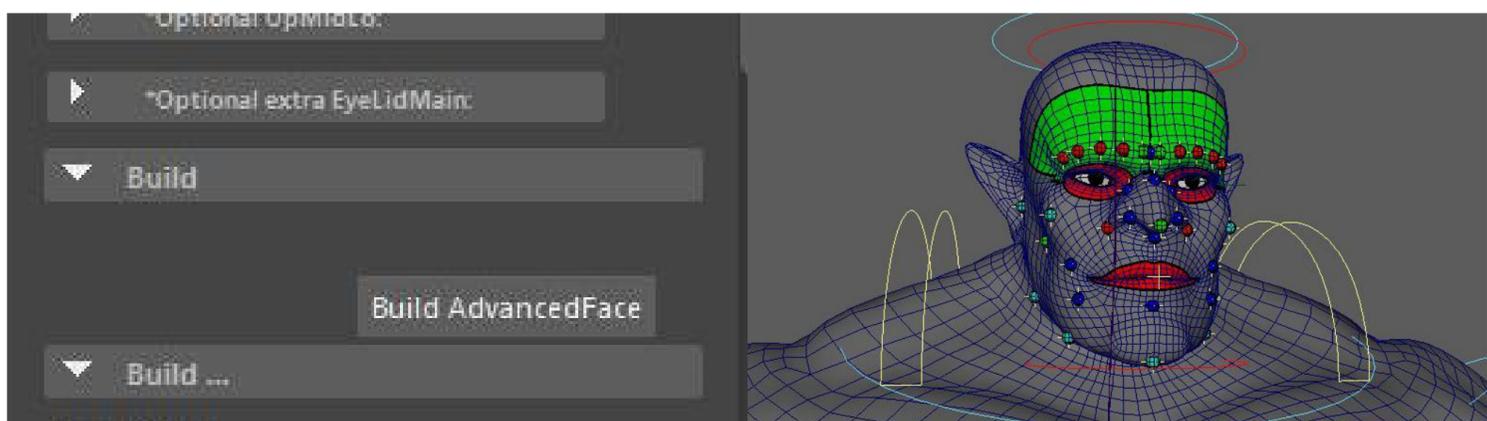


Figura 28. Interfaz de Advanced Skeleton. Rig facial de Noboru. Elaboración propia (2024).

Al finalizar el proceso de construcción del rig facial, tendrás un personaje con su *rig* corporal y facial completos. Aprovecha este momento para revisar y perfeccionar los pesos de las deformaciones, utilizando herramientas como *Paint Skin Weights*, *Hammer Tool* y *Smooth Weights* si es necesario.

Puedes mejorar la forma de los *Blend Shapes* utilizando el *Shape Editor* en Autodesk Maya. Esto te permitirá afinar los detalles de las expresiones faciales y lograr un mayor nivel de realismo o estilización según las necesidades de tu proyecto.



Figura 29. Rig de Noboru. Rig realizado en Autodesk Maya con Advanced Skeleton. Elaboración propia (2024).

Una vez satisfecho con los resultados del rig corporal y facial, procede a exportar tu personaje. Asegúrate de seguir los procedimientos adecuados para la exportación de *rigs* y *Blend Shapes* para mantener la compatibilidad con Unreal Engine 5.

5. Exportación e importación a *Unreal Engine*

En esta sección, se explicará el proceso de exportación del rig creado en Advanced Skeleton y su importación en Unreal Engine 5. Este proceso incluye la configuración necesaria para garantizar que tanto el rig corporal como el facial funcionen correctamente en el entorno del motor de juego.

Exportación desde *Advanced Skeleton*

- **Exportación FBX:** Dirígete al apartado “*Export*” en Advanced Skeleton. A pesar de que la opción “*Manequinn Export*” tiene compatibilidad con la recién incorporada función de “*Modular Control Rig*”, esta no es útil para esta investigación. Actualmente, el “*Modular ControlRig*” no reconoce la estructura del rig facial de Advanced Skeleton. Por lo tanto, la mejor alternativa es proceder con la opción de “*FBX Export*” y, adicionalmente, “*Control RigExport*”.



Figura 30. Interfaz de Advanced Skeleton. Captura de pantalla de opciones de exportación. Elaboración propia (2024).

- **Preparación del Personaje:** Si necesitas tener múltiples *Skeletal Meshes* dentro de un *Blueprint* para controlar diferentes propiedades de las prendas de ropa de tu personaje de forma individual, puedes importar el FBX en Blender. Esto permite mayor comodidad y evita dañar el rig ya construido en Advanced Skeleton.

Manipulación en Blender

- **Exportación de *Skeletal Meshes*:** En Blender, exporta las diferentes prendas de ropa que necesitas en cada *Skeletal Mesh*, siempre manteniendo seleccionado el rig. En las propiedades de exportación de Blender, asegúrate de seleccionar las opciones “*Mesh*” y “*Armature*”, y de desmarcar la opción “*Add Leaf Bones*”.

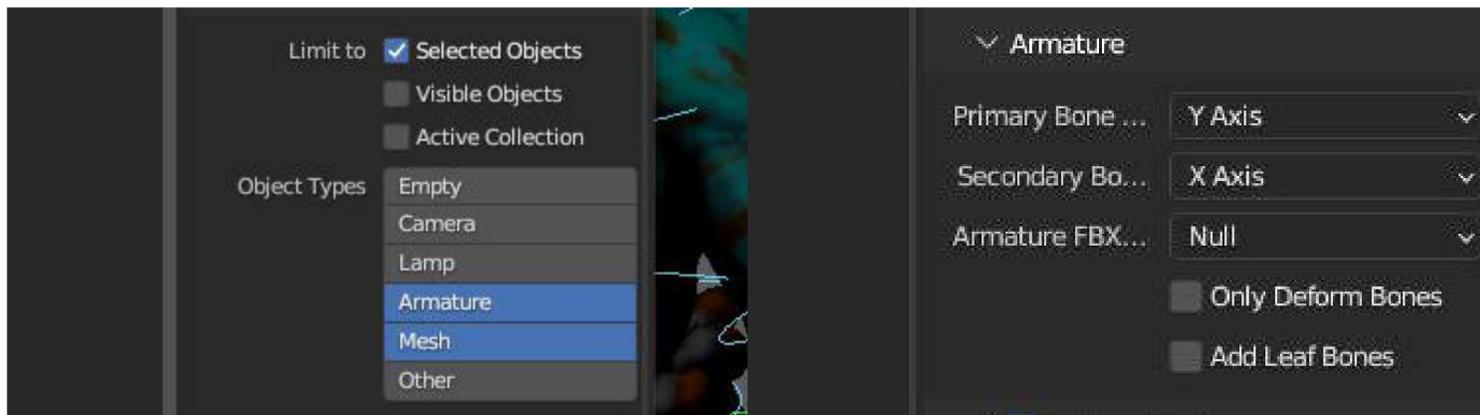


Figura 31. Exportación de rig. Captura de pantalla de opciones de exportación de FBX. Elaboración propia (2024).

Importación a *Unreal Engine*

- **Importación de *FBX*:** Al importar los múltiples *FBX* a Unreal Engine, selecciona en “*Skeleton*” el *rig* principal, de lo contrario, habrá múltiples copias de un mismo *rig*.

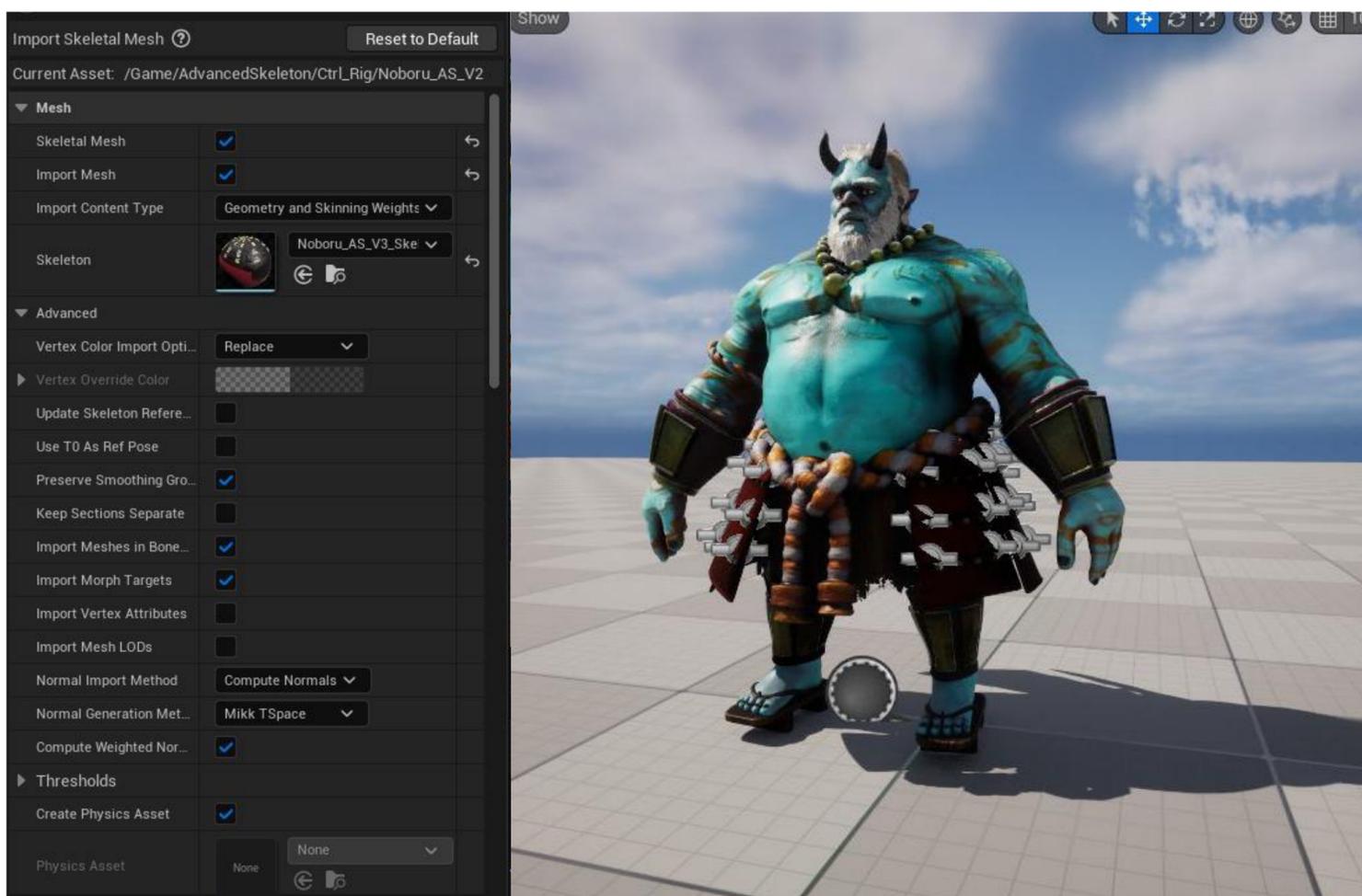


Figura 32. Noboru en Unreal Engine 5. Modelo importado a Unreal Engine. Elaboración propia (2024).

- **Material Graph Editor:** Dirígete al *Material Graph Editor* para construir la estructura de nodos de los materiales del personaje y asignarlos en cada *Skeletal Mesh*. Esto es determinante para mantener la apariencia visual del personaje en el motor de juego.

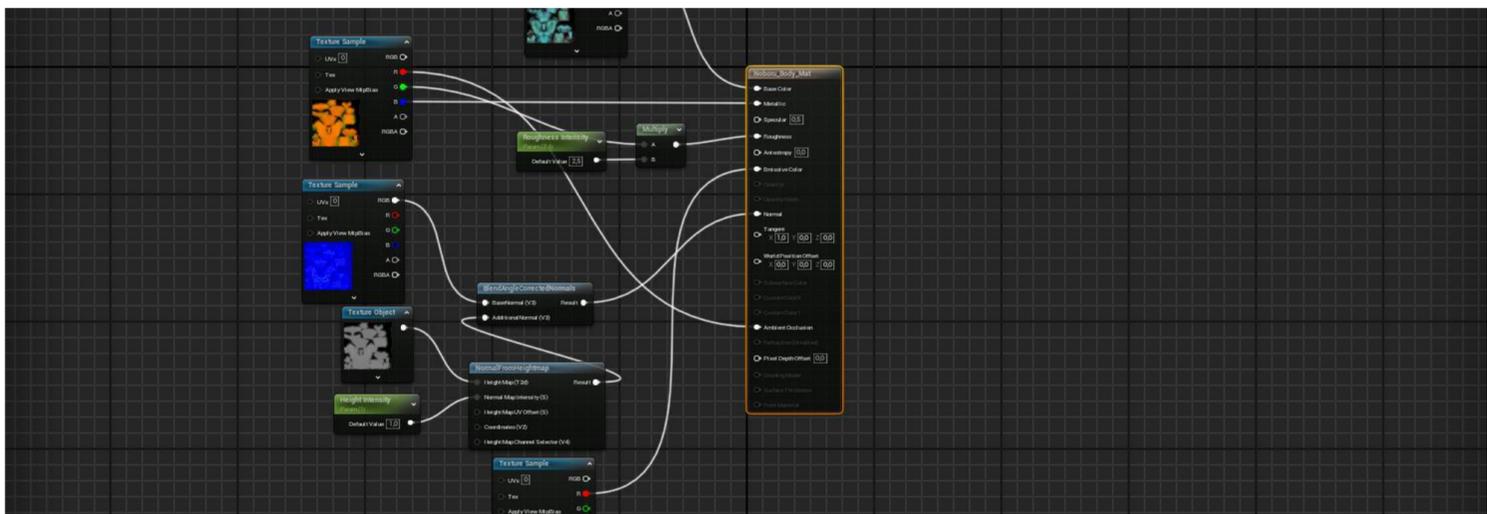


Figura 33. Editor de materiales. Captura de pantalla de nodos del material. Elaboración propia (2024).

- **Asignación en Blueprint:** Asigna la jerarquía de los *Skeletal Meshes* dentro del *Blueprint*, asegurándote de que cada pieza de ropa y el cuerpo principal estén correctamente organizados. Es bastante importante considerar que todos los *Skeletal Meshes* que sean hijos en la jerarquía deben ser asignados y emparentados desde el *Construction Script* al componente padre a través del nodo “*Set Leader Pose Component*”.

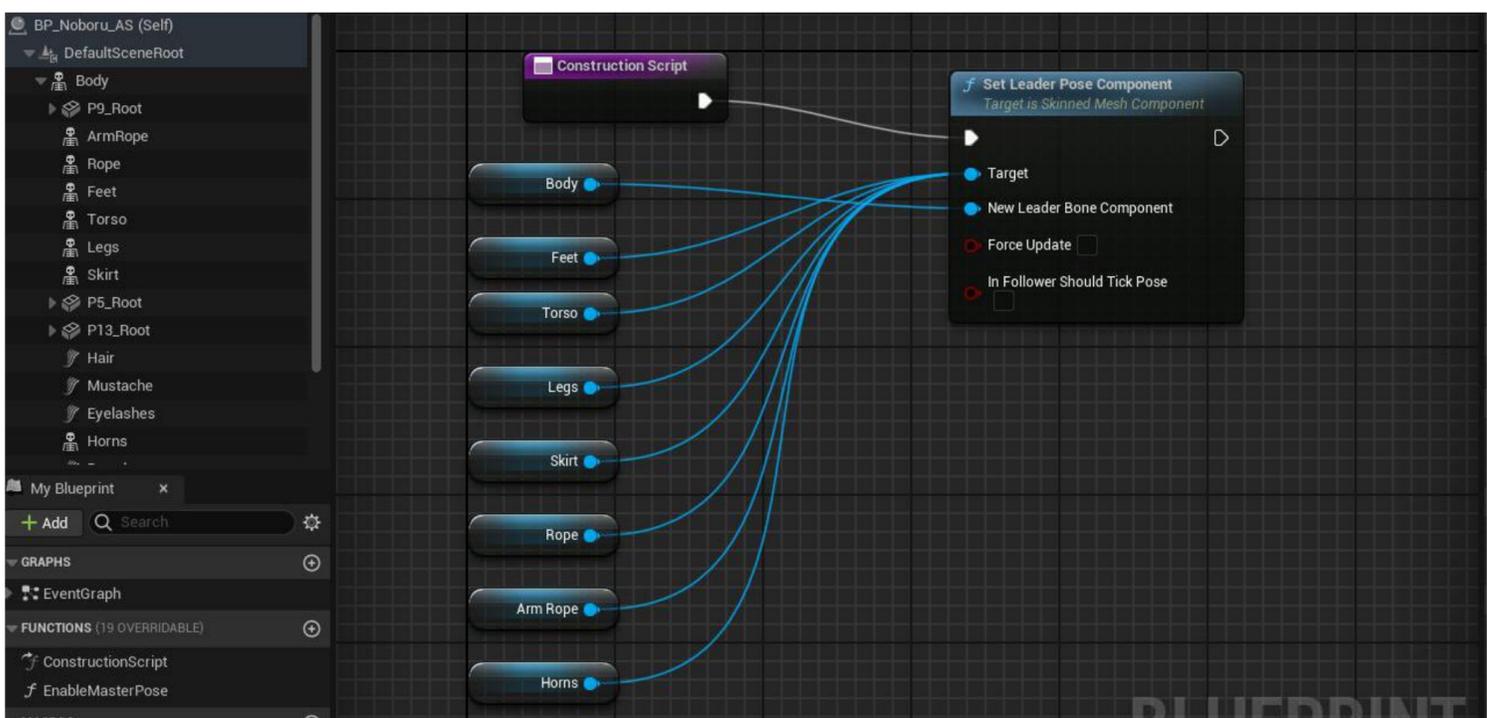


Figura 34. Construction Script. Captura de pantalla de nodos del Construction script. Elaboración propia (2024).

- **Creación del Control Rig:** Selecciona el *Skeletal Mesh* principal dentro del Content Browser y presiona el botón "Create - Control Rig". Asigna el rig de *Advanced Skeleton* y dirígete a "Tools - Execute Python Script". Esto creará el *control rig* necesario para tu personaje.

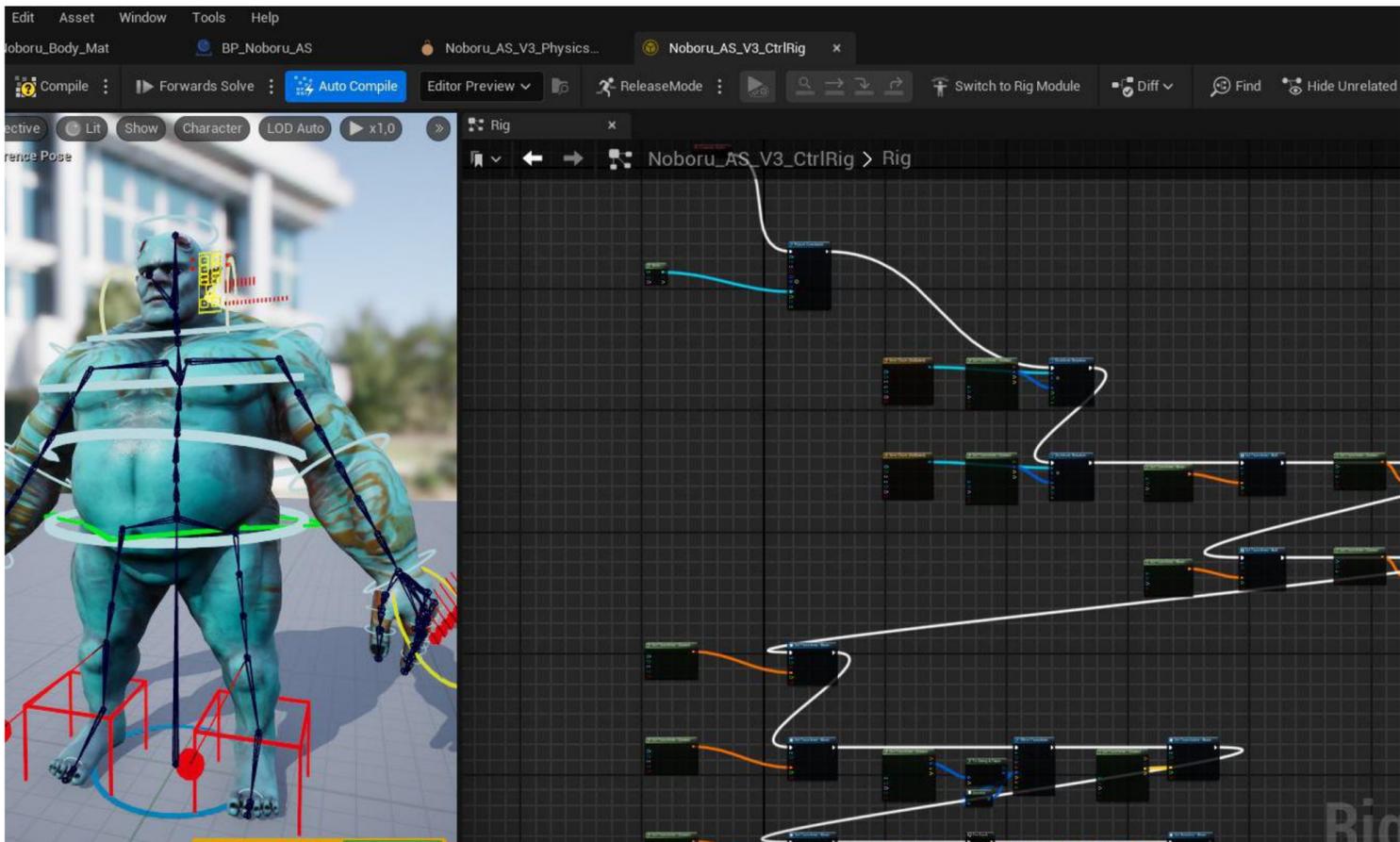


Figura 35. Control Rig. Captura de pantalla del control rig en UE5. Elaboración propia (2024).

- **Asignación del Control Rig:** Al asignar el recién creado control rig dentro del *Skeletal Mesh* principal (*Body*), cada vez que el personaje sea instanciado en un *Level Sequence*, podrás utilizar los controladores tanto corporales como faciales de tu personaje.

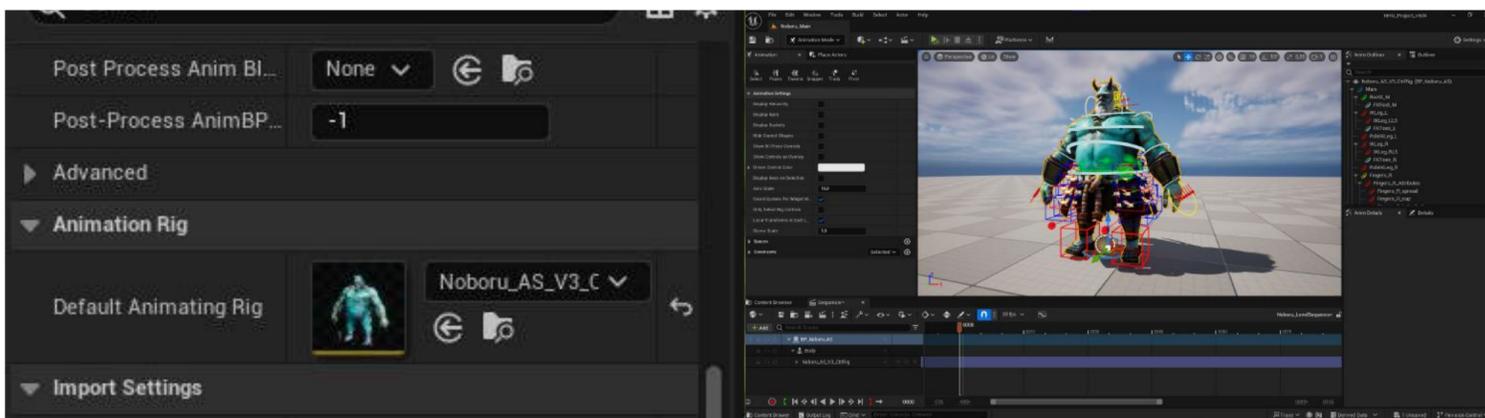


Figura 36. Level Sequence. Captura de pantalla del Level Sequence de Noboru. Elaboración propia (2024).

Ahora, únicamente se necesita preparar las dinámicas y simulaciones del personaje, un rubro que será abordado, tanto para MetaHuman como para Advanced Skeleton, en el capítulo 5.

RIGGING EN METAHUMAN

1. Introducción

En los últimos años, MetaHuman ha experimentado una evolución significativa, posicionándose como una herramienta clave en la creación de personajes realistas en la industria de la animación y los videojuegos. Este avance ha llevado a su implementación en diversos proyectos AAA, tales como *Hellblade 2*, *Matrix Awakens* y *Death Stranding 2*. Estos ejemplos no solo demuestran la versatilidad y el poder de MetaHuman, sino también su capacidad para elevar los estándares de realismo y detalle en los personajes digitales.

A pesar de que, en esta documentación, se considera a MetaHuman como un software comparable con una herramienta de auto-rig, es crucial entender que su flujo de trabajo es distintivo. A diferencia de las herramientas tradicionales de *auto-rigging*, MetaHuman se basa en una infraestructura compleja que combina la precisión del *rigging* automático con una capacidad de personalización profunda a través de la ingeniería inversa.

En este capítulo, se explorará cómo se puede aprovechar MetaHuman para elaborar un personaje personalizado, desafiando sus límites tradicionales. En este sentido, se analizarán los procedimientos y técnicas necesarios para adaptar y modificar MetaHuman, desde la configuración inicial hasta la implementación final en un motor de juego. A lo largo de esta documentación, se proporcionará una visión detallada de cómo desglosar y reconstruir MetaHuman, lo que permitirá a los desarrolladores y artistas crear personajes únicos que se ajusten a sus necesidades específicas.

Al final de este capítulo, se espera demostrar que MetaHuman no es solo una herramienta para crear avatares hiperrealistas, sino también una plataforma flexible y poderosa que puede integrarse y adaptarse a una amplia gama de estilos y requisitos en la producción de videojuegos y animación.

2. *Wrapping* de la malla

Para adaptar cualquier personaje a MetaHuman, es indispensable someterlo al proceso de *Wrapping*. MetaHuman no puede adaptar cualquier malla a su rig, por lo que es esencial respetar la malla original de esta herramienta, así como el índice de cada uno de sus vértices. Esto significa que MetaHuman no puede ser sometido a ningún proceso que altere la cantidad de polígonos, únicamente el posicionamiento de estos.

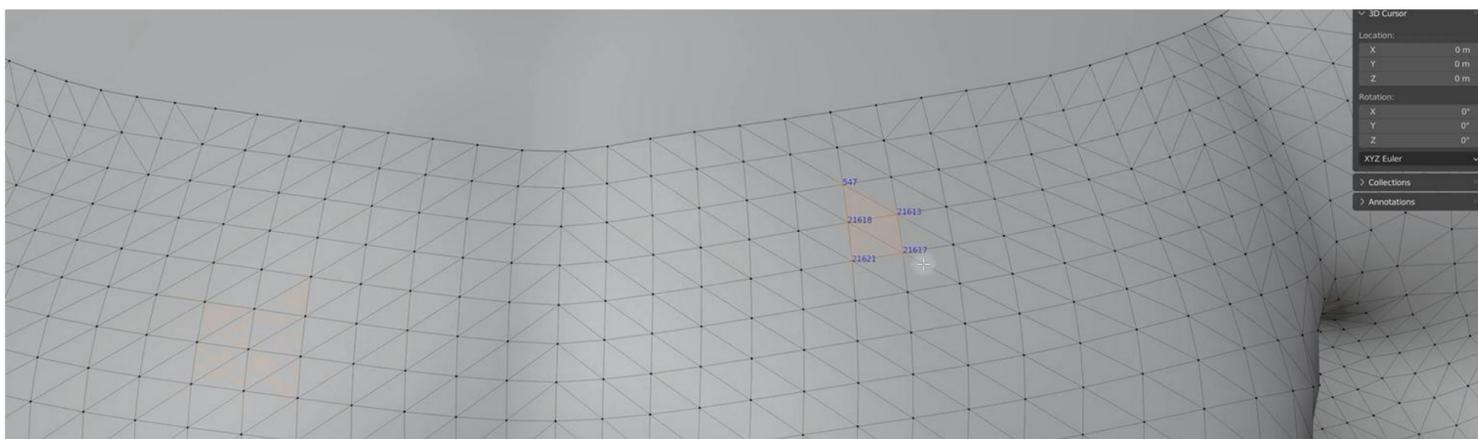


Figura 37. Índice de vértices. Captura de pantalla del vértices de MetaHuman. Elaboración propia (2024).

El primer paso es acceder a MetaHuman Creator y crear un personaje lo más cercano posible a su versión personaje estilizada. Es probable que exista una diferencia significativa entre ambos, pero esto no representa un problema. Una vez que el MetaHuman esté finalizado, se importará a Unreal Engine a través de la propia galería de *Quixel Bridge*.

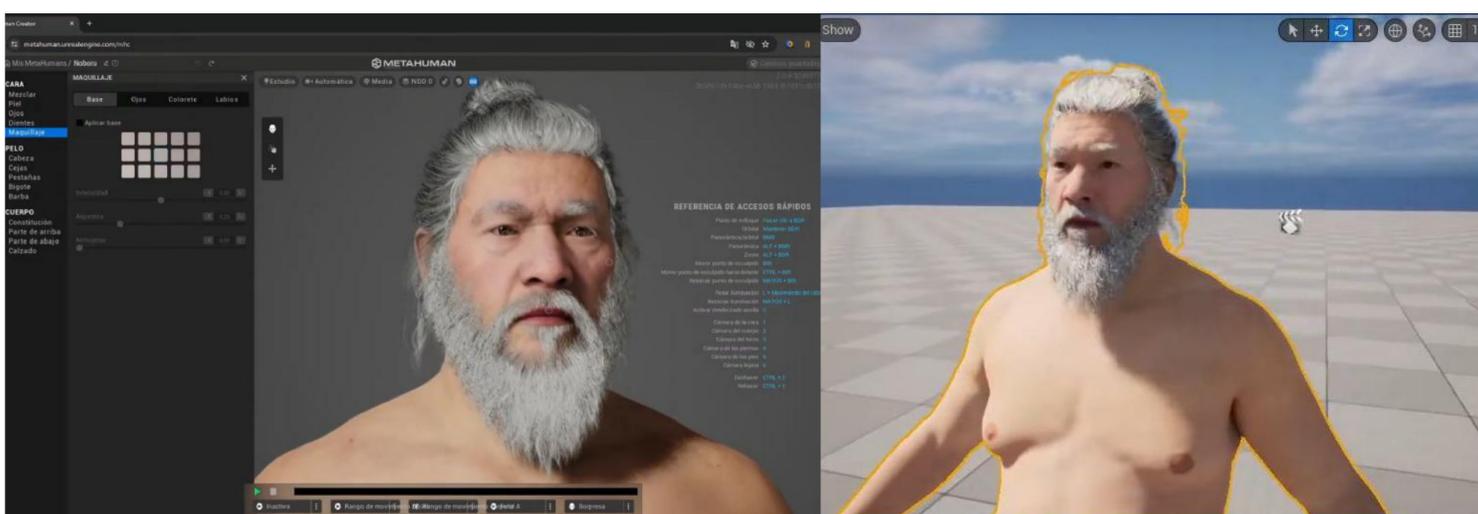


Figura 38. MetaHuman Creator. Captura de pantalla del creador de MetaHumanos. Elaboración propia (2024).

Al agregar el *metahuman* a este proyecto de Unreal Engine, se podrá arrastrar su *Blueprint* al propio nivel. Para exportarlo e importarlo a cualquier otro software, se arrastra el *Blueprint* al *Level Sequence* y, luego, se hace clic derecho y se selecciona “*Export*”. De esta manera, se obtendrá un archivo FBX que contendrá toda la estructura del propio *metahuman*.

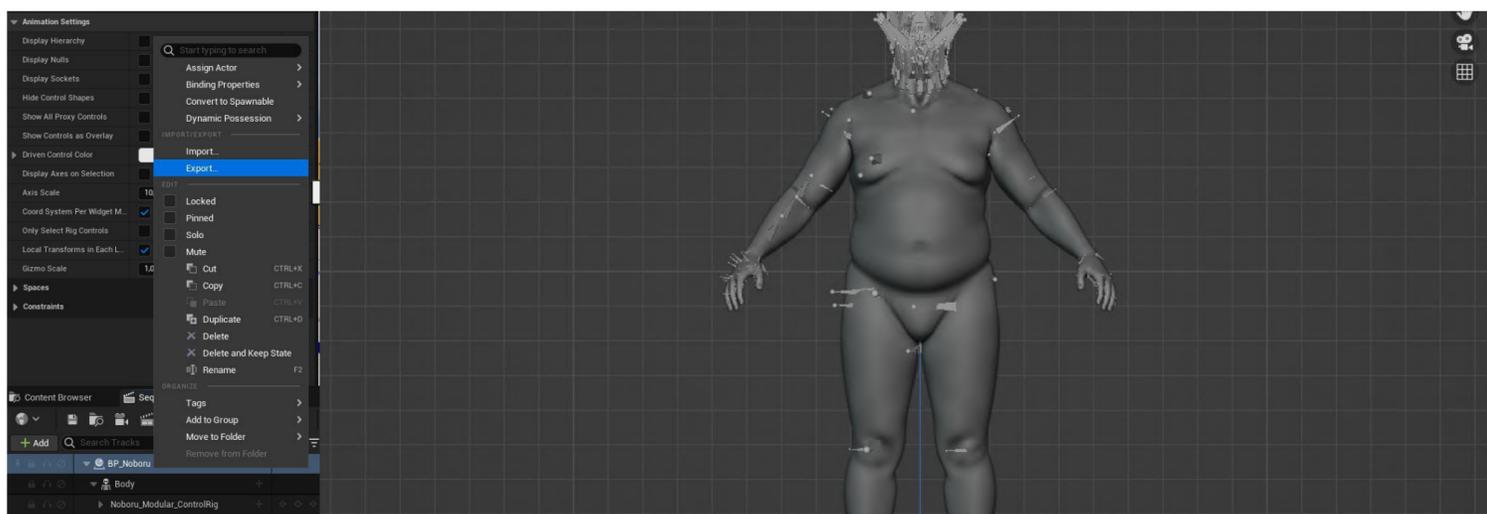


Figura 39. Rig en Blender. Captura de pantalla de Metahuman importado en Blender. Elaboración propia (2024).

En este punto, se podría trabajar con el modelo de múltiples formas. Sería posible importarlo a Maya o Blender y construir un *Blend Shape* desde cero, esculpiendo a este *metahuman* y modelándolo desde cero. Sin embargo, esta opción podría no ser eficiente en una producción que ya pasó por el proceso de modelado y texturizado. En caso de no contar aún con un personaje finalizado, vale la pena considerar crear el personaje esculpiendo un *Blend Shape* del *metahuman*.



Figura 40. Blendshapes de Metahuman. Captura de pantalla de la transformación corporal de Metahuman. Elaboración propia (2024).

Si ya se cuenta con un personaje finalizado, se selecciona el modelo *High Poly* de este personaje y el propio *metahuman*, y ambos se importan a ZBrush. Se organizan ambos modelos como dos *Subtools* en ZBrush, siendo el primero el *metahuman* y el segundo el modelo terminado.

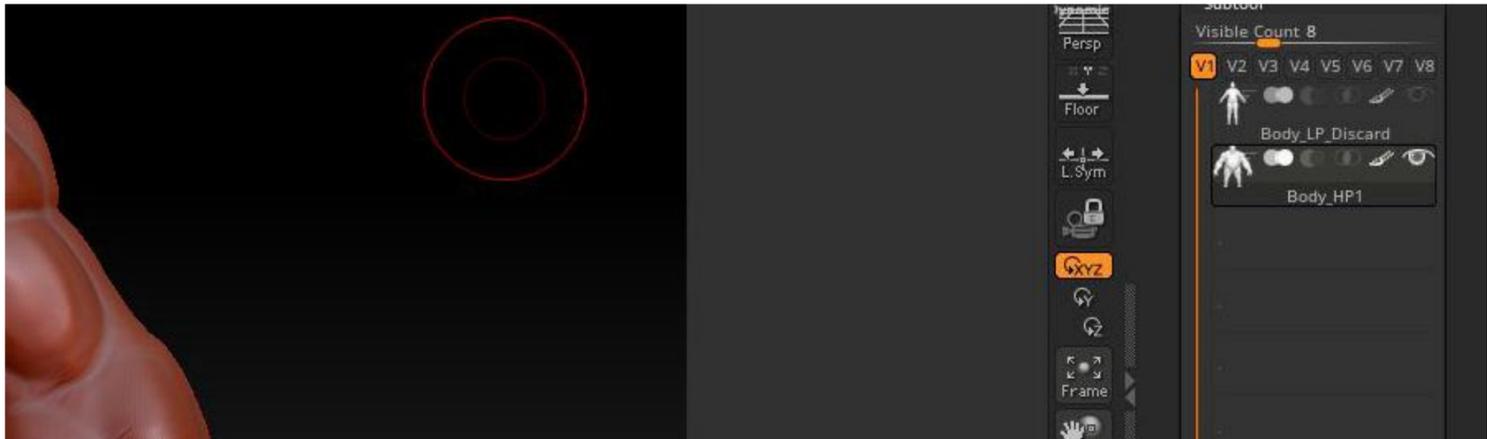


Figura 41. Subtools en Zbrush. Captura de pantalla de modelos en Zbrush. Elaboración propia (2024).

A continuación, se utilizará la licencia estudiantil del *Plugin ZWrap* para continuar con el proceso de *Wrapping*. Esta licencia puede ser solicitada en el sitio web oficial del desarrollador:

<https://www.russian3dscanner.com/zwrap/>

Una vez instalado el plugin de *ZWrap*, se procede a abrirlo. Se deben marcar puntos clave en los personajes que permitan trasladar correctamente los vértices del *metahuman* a la estructura anatómica del personaje estilizado. Al finalizar este proceso, se hace clic en el apartado de *Wrapping* y se presiona el botón "*Start Wrapping*". Este proceso puede tardar unos minutos, pero si todos los puntos fueron colocados correctamente, se podrá ver al modelo de *metahuman* convertido en el personaje creado.

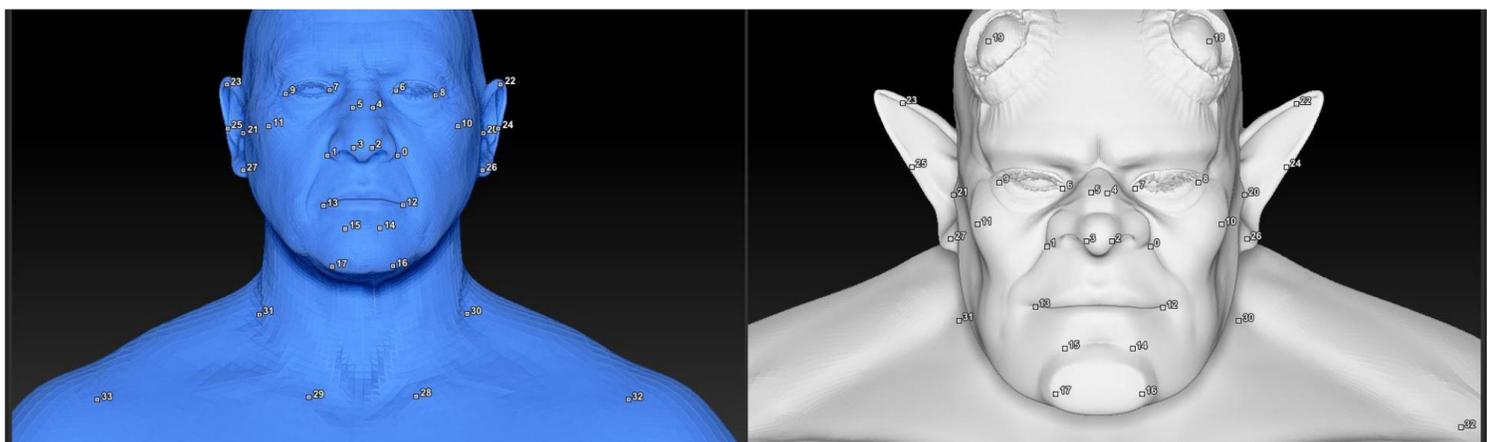


Figura 42. Interfaz de ZWrap. Captura de pantalla de wrapping de Noboru. Elaboración propia (2024).

Cabe señalar que, aunque ZWrap es una herramienta bastante eficiente, en ocasiones puede ser necesario ajustar ligeramente la malla a través de las herramientas de *retopología* de Maya para mejorar el posicionamiento de los polígonos. En el caso de este proyecto, esto no fue necesario.

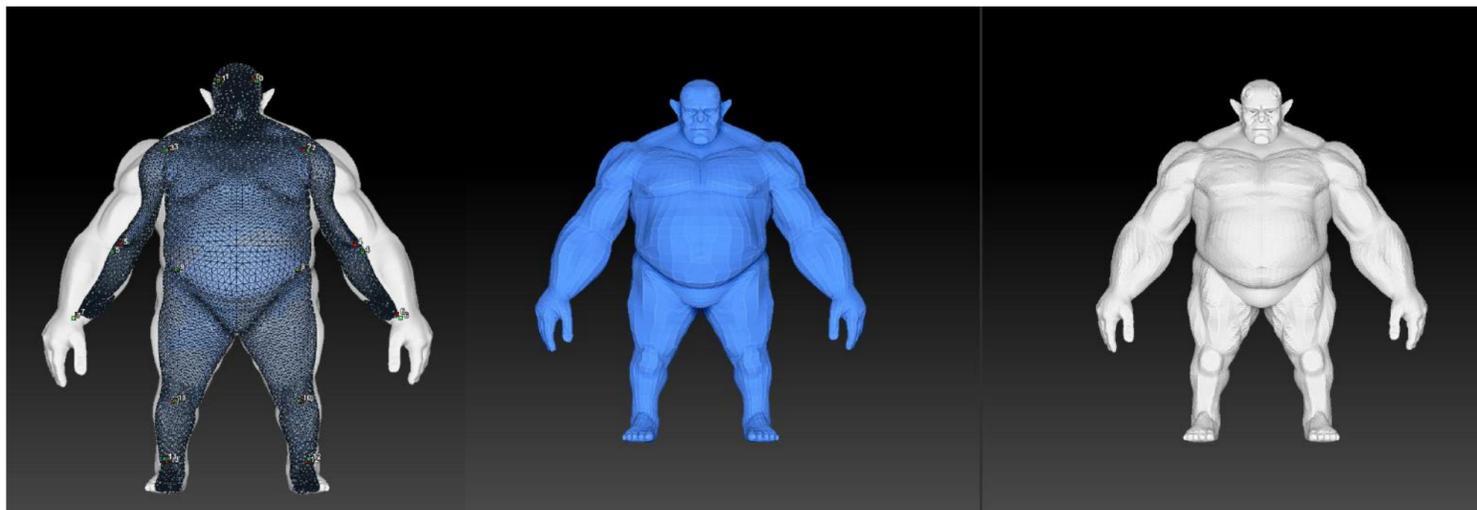


Figura 43. Interfaz de ZWrap. Captura de pantalla de wrapping de Noboru. Elaboración propia (2024).

El modelo creado puede ser exportado y estará completamente listo para ser importado más tarde dentro de MetaHuman. Este tema será abordado en el apartado de exportación e importación a Unreal Engine.

3. Proceso de *Baking* de texturas

En cuanto se tiene el modelo 3D listo, no es viable para ninguna producción comenzar de cero el proceso de texturizado en personajes que ya estaban terminados. Para esto, es posible someterlos al proceso de *Baking*. Este es un procedimiento en el que se trasladan las texturas que posee el personaje original y se proyectan sobre una textura vacía en el UV del propio *metahuman*. Este proceso se replicará con cada uno de los mapas de textura del personaje.

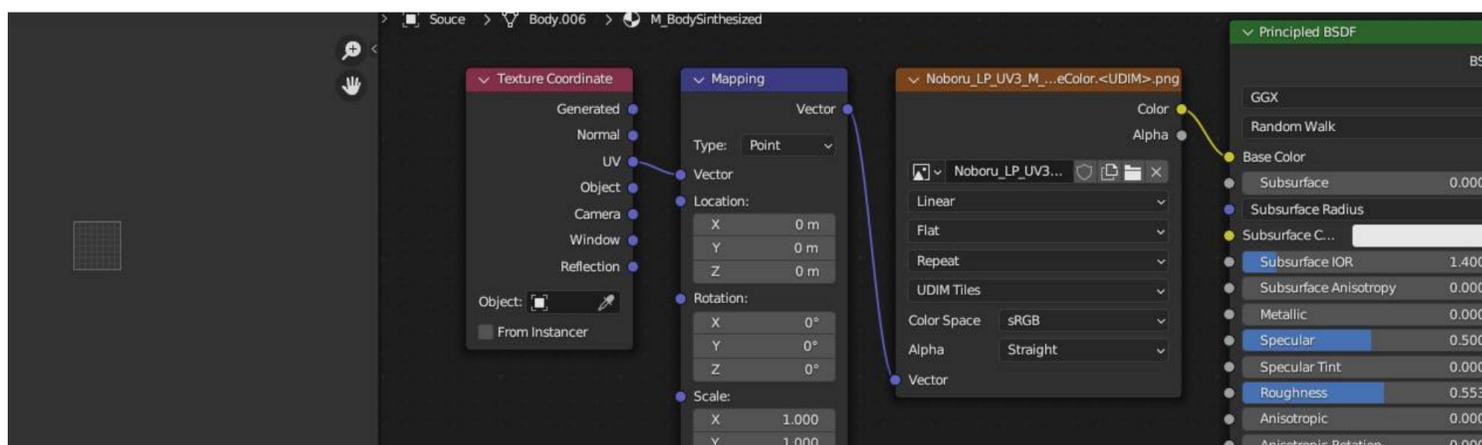


Figura 44. Shading para Baking. Viewport de Shading en Blender. Elaboración propia (2024).

Para comenzar, es recomendable crear un archivo limpio en Blender, en el cual se tendrán tres objetos: el *Mesh* original (*Source*), el *Mesh* de MetaHuman (*Target*), y un duplicado del *Mesh* de MetaHuman, al cual se le ha aplicado el parámetro “*inflate*” en las herramientas de esculpido (*Cage*). Posteriormente, se creará un material simple en los propios *Meshes*. En el *Source*, se colocará, en el Base Color, la textura que se quiere trasladar y, en el *Target*, se incorporará una textura vacía con el nombre del mapa de textura que se busca generar.

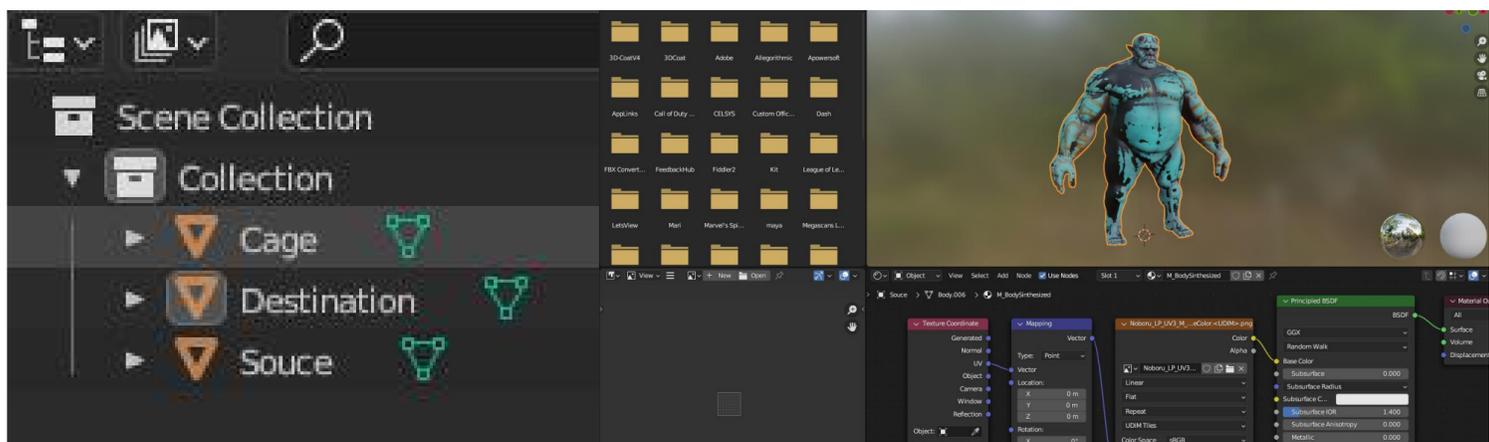


Figura 45. Bake de texturas. Proceso de Baking de Noboru. Elaboración propia (2024).

Posteriormente, es necesario dirigirse a las opciones de Render, donde se seleccionará el motor de render *Cycles* y se bajará hasta la opción “*Bake*”. En ella, únicamente se marcarán las casillas “*Color*”, “*Selected to Active*” y “*Cage*”, y se debe elegir el propio *Cage object*. Luego, se presionará el botón “*Bake*”.

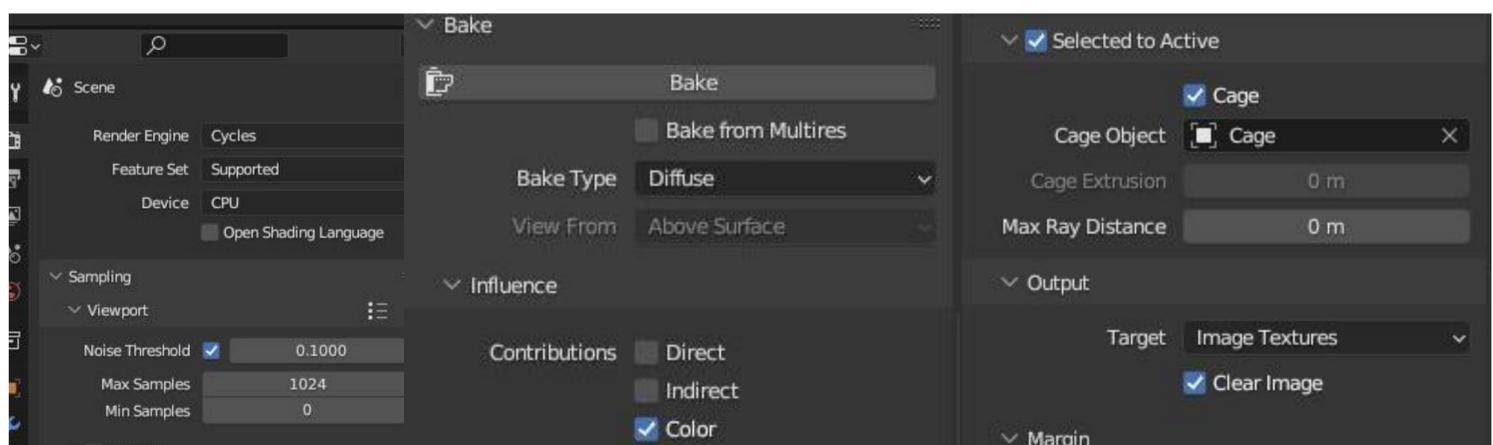


Figura 46. Bake de texturas. Proceso de Baking de Noboru. Elaboración propia (2024).

Al finalizar, en el visor de imágenes de Blender, se podrá observar cómo se ha trasladado la textura a los UV del propio modelo de MetaHuman. Se abrirá el menú, se seleccionará *Image - Save As* y se guardará el nuevo mapa de texturas. Este proceso se debe repetir con cada uno de los mapas de textura.



Figura 47. Bake de texturas. Resultado del baking de texturas. Elaboración propia (2024).

Si se desea hacer mejoras, estos mapas pueden ser importados como texturas en un Fill Material de Substance Painter. Allí, se pueden hacer los ajustes necesarios sobre este material y reexportarlo.

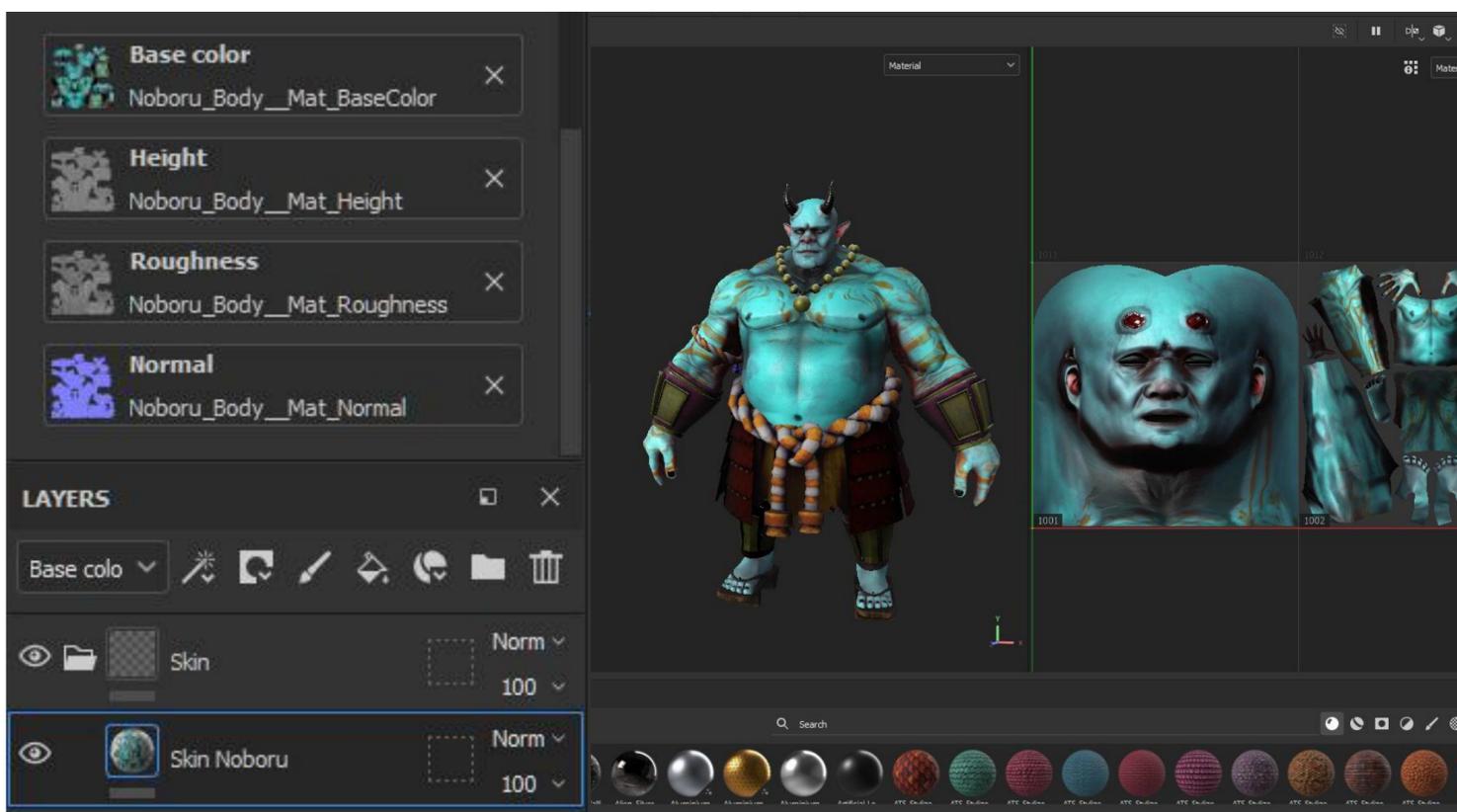


Figura 48. Proceso de texturizado. Resultado del texturizado en Substance Painter. Elaboración propia (2024).

Es importante destacar que, en esta etapa del proceso, puede ser necesario realizar una integración de los mapas de texturas de Base Color y normales del *Face Mesh* de MetaHuman. Esto es indispensable para no perder todos los detalles que otorgan las arrugas a la estructura del rig, tal como se explicó anteriormente. Para este proceso, es necesario realizar una integración a través de Substance Painter y reexportar las texturas modificadas y adaptadas para MetaHuman y su uso dentro de Unreal Engine.

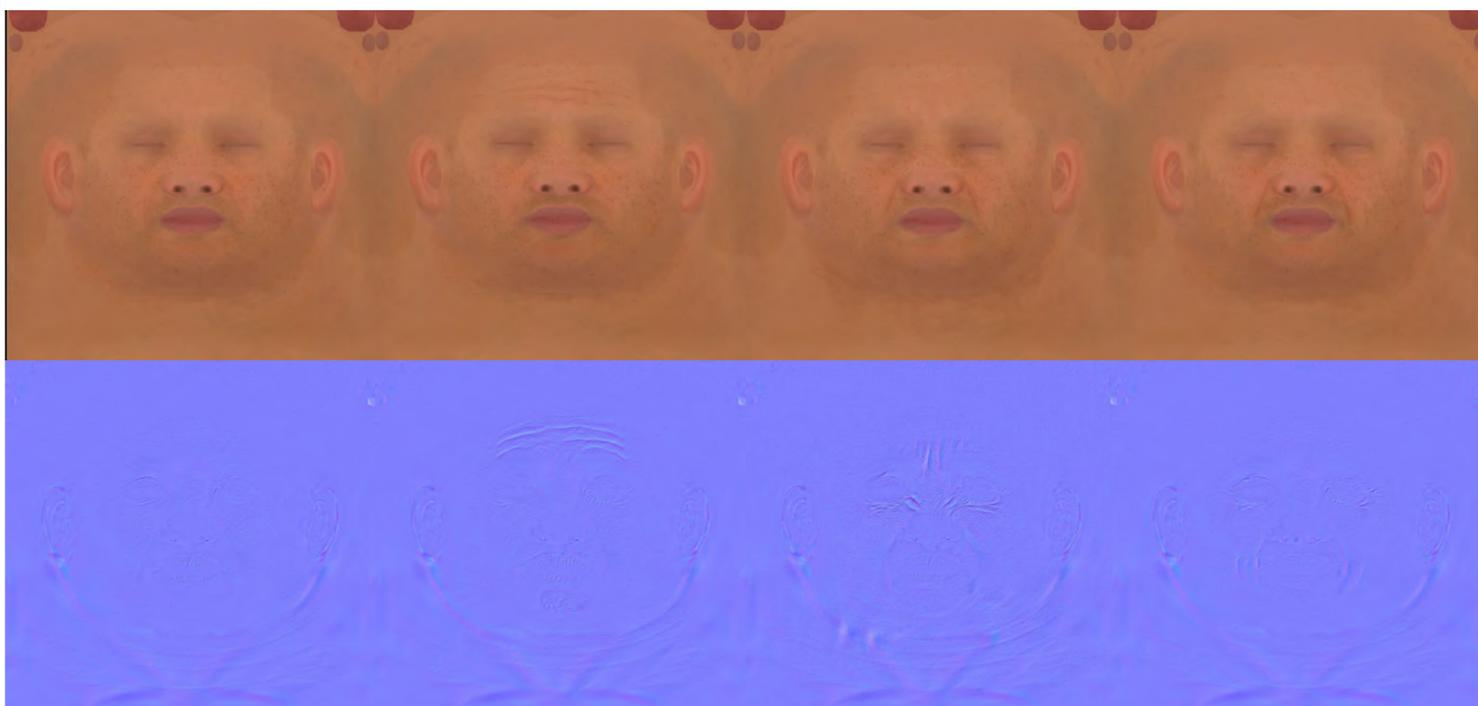


Figura 49. Comparación de texturas base. Texturas de color y normales de Metahuman. Elaboración propia (2024).



Figura 50. Comparación de texturas base. Texturas de color y normales de Metahuman aplicadas a Noboru. Elaboración propia (2024).

También es importante modificar la textura *Under Wear Mask* del material base de MetaHuman dentro de Adobe Photoshop. De lo contrario, el personaje siempre tendrá unos bóxeres añadidos sobre el cuerpo base, lo cual puede no ser el resultado deseado para ciertos personajes. Para evitarlo, solo se debe eliminar todo lo demarcado con color magenta.

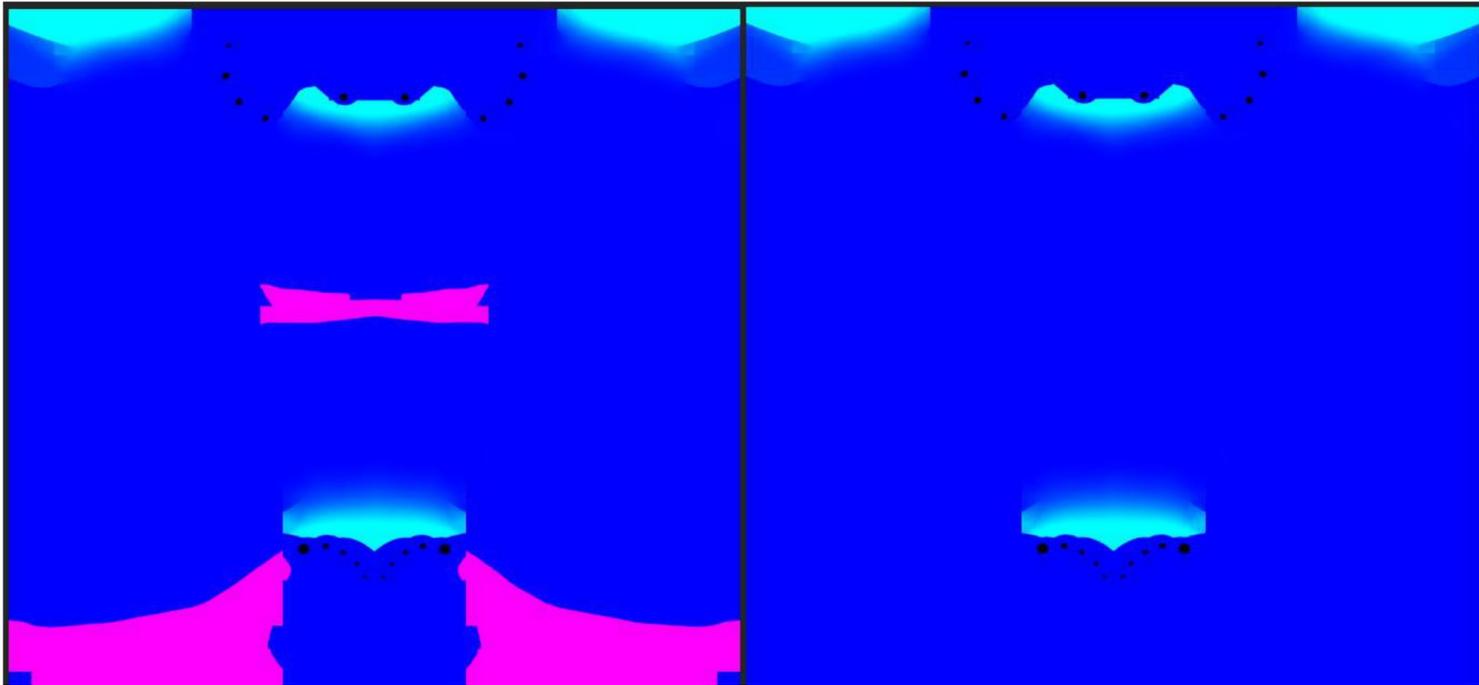


Figura 51. Under Wear Mask. Edición de la textura Under Wear Mask. Elaboración propia (2024).

Con estos pasos, este modelo 3D estará completamente texturizado y listo para ser importado y utilizado en MetaHuman dentro de Unreal Engine, asegurando que mantenga la calidad visual y los detalles necesarios para cualquier producción de alto nivel.

4. Pintado de pesos

En el caso de las prendas de ropa, no es necesario someterlas al proceso de *Wrapping* como se hace con el cuerpo del personaje. Estas vestimentas pueden ser reutilizadas con el rig de MetaHuman directamente. Sin embargo, es completamente necesario importarlas junto con el esqueleto del *metahuman* en un proyecto de Maya o Blender para unir los objetos al esqueleto y proceder con el proceso de pintado de pesos. Este proceso debe realizarse con bastante cuidado e implica la inversión de cierto tiempo para garantizar una correcta deformación y adaptación de la ropa al *rig*.

Para comenzar, importa las prendas de ropa y el esqueleto del *metahuman* a tu software de elección, ya sea Maya o Blender. Asegúrese de que las prendas estén correctamente alineadas y posicionadas con respecto al esqueleto. Una vez que las vestimentas y el esqueleto estén en su lugar, selecciona las prendas y asócialas al esqueleto utilizando la opción de *Skinning* disponible en tu software.

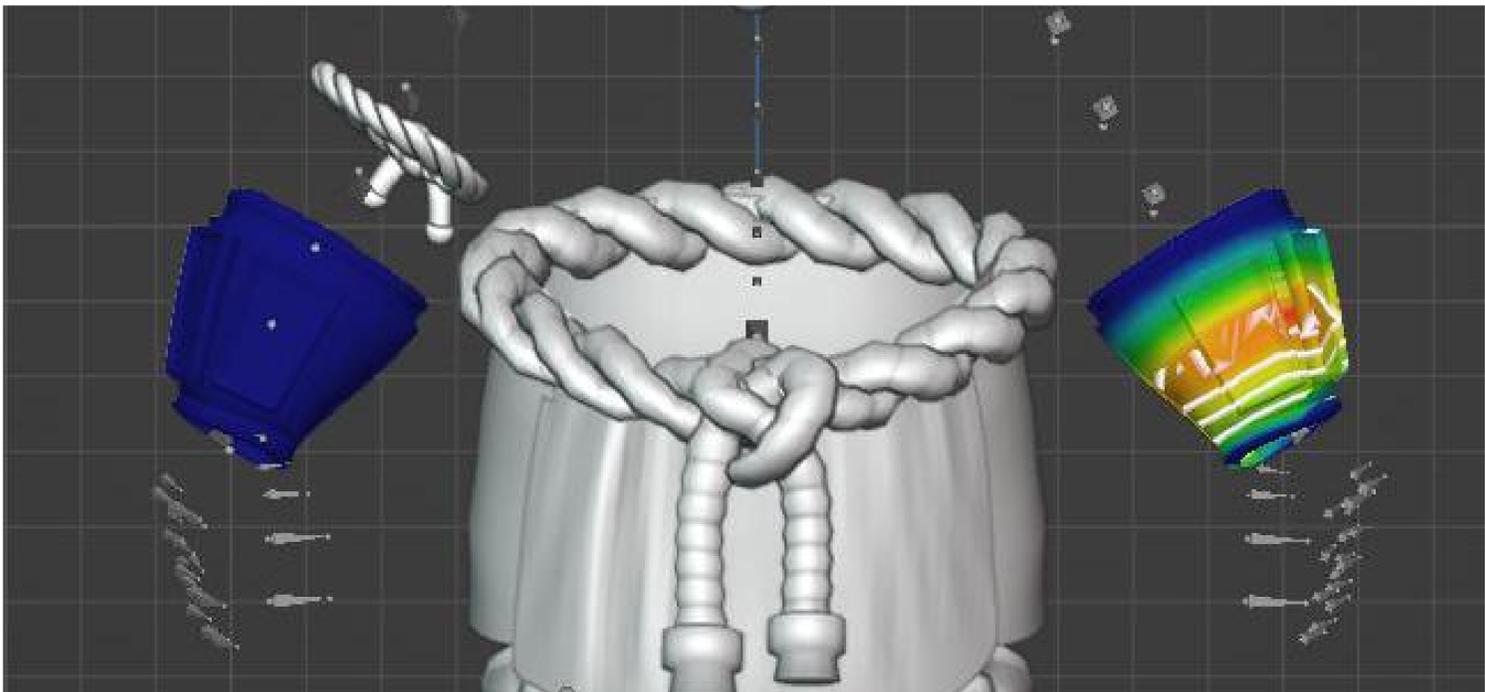


Figura 52. Pintado de pesos. Ajuste de pesos en ropa. Elaboración propia (2024).

El proceso de pintado de pesos es fundamental para garantizar que las prendas de ropa se deformen correctamente junto con el cuerpo del personaje. Utiliza las herramientas de pintura de pesos para asignar correctamente la influencia de cada *joint* sobre la malla de las prendas. Este proceso puede ser laborioso y requiere paciencia para confirmar que las deformaciones sean naturales y sin artefactos.

Es importante destacar que, para las prendas de ropa que requieran de simulaciones de tela o físicas, aunque es esencial pintar de forma correcta sus pesos, no es necesario que estos sean perfectos. Las simulaciones de este tipo aplicadas dentro de un motor de juego, como Unreal Engine, pueden variar significativamente la forma en que las prendas se comportan. Por lo tanto, el pintado de pesos debe enfocarse en proporcionar una base sólida que permita a las simulaciones actuar de manera realista.

Al finalizar el pintado de pesos, es importante asegurarse de probar las deformaciones del esqueleto y ajustar cualquier área problemática. Luego de que se esté satisfecho con los resultados, las prendas de ropa estarán listas para ser exportadas e importadas a Unreal Engine, donde se les aplicarán las simulaciones de tela o físicas necesarias para el proyecto.

Este enfoque garantizará que las vestimentas no solo se adapten correctamente al rig de MetaHuman, sino que también respondan de manera adecuada a las simulaciones de física, proporcionando un resultado visualmente coherente y realista en el entorno del motor de juego.

5. Exportación e Importación a *Unreal Engine*

Este apartado es uno de los más complejos y extensos de toda la documentación, por lo que se segmentará en varias etapas para facilitar su comprensión. Para la mayor parte de este proceso, será necesario utilizar el plugin *Mesh Morpher*. Asimismo, para la mayoría de los personajes, la versión Lite puede ser suficiente, pero en el caso de este personaje, se utilizó la versión de paga mensual, ya que se requirió la herramienta *Poser* para una etapa del proceso.

Este *plug-in* puede ser adquirido desde el *website* oficial del desarrollador: <https://www.meshmorpher.com/plans-pricing>

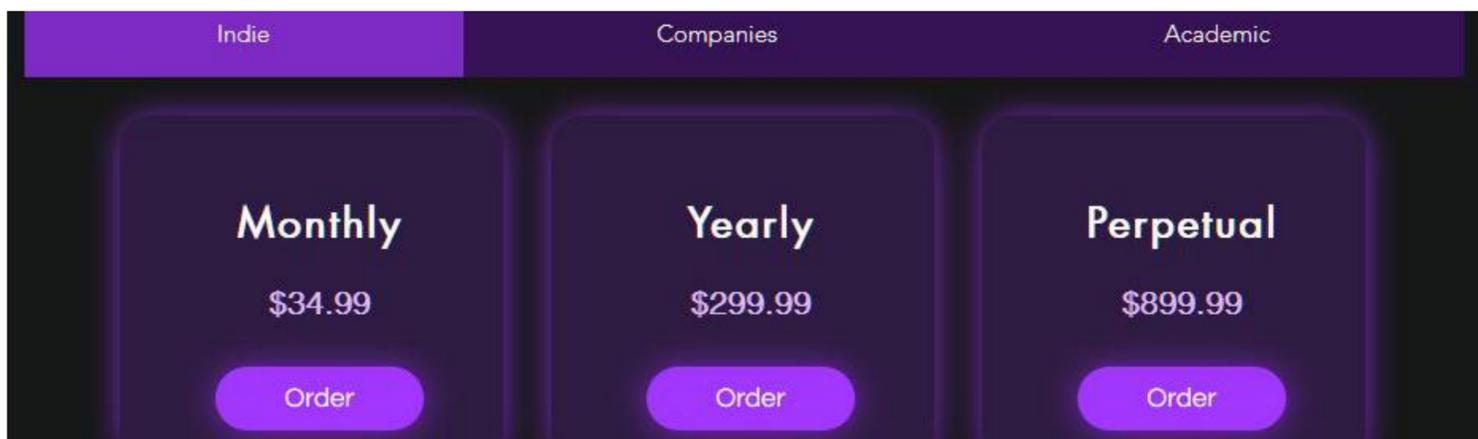


Figura 53. Precios de Mesh Morpher. Captura de pantalla de precios del sitio web de Mesh Morpher. Recuperado del website de Mesh Morpher (2024).

Etapa A: Creación del *Morph Target* y materiales

Lo primero que se debe hacer es convertir el cuerpo base de este *metahuman* en el personaje estilizado. Para esto, se seleccionará el *Skeletal Mesh* del cuerpo y/o la cabeza del *metahuman* y se presionará el acceso directo de *Mesh Morpher*. En el plugin, se dará clic al botón “*Create from Mesh Files*”.

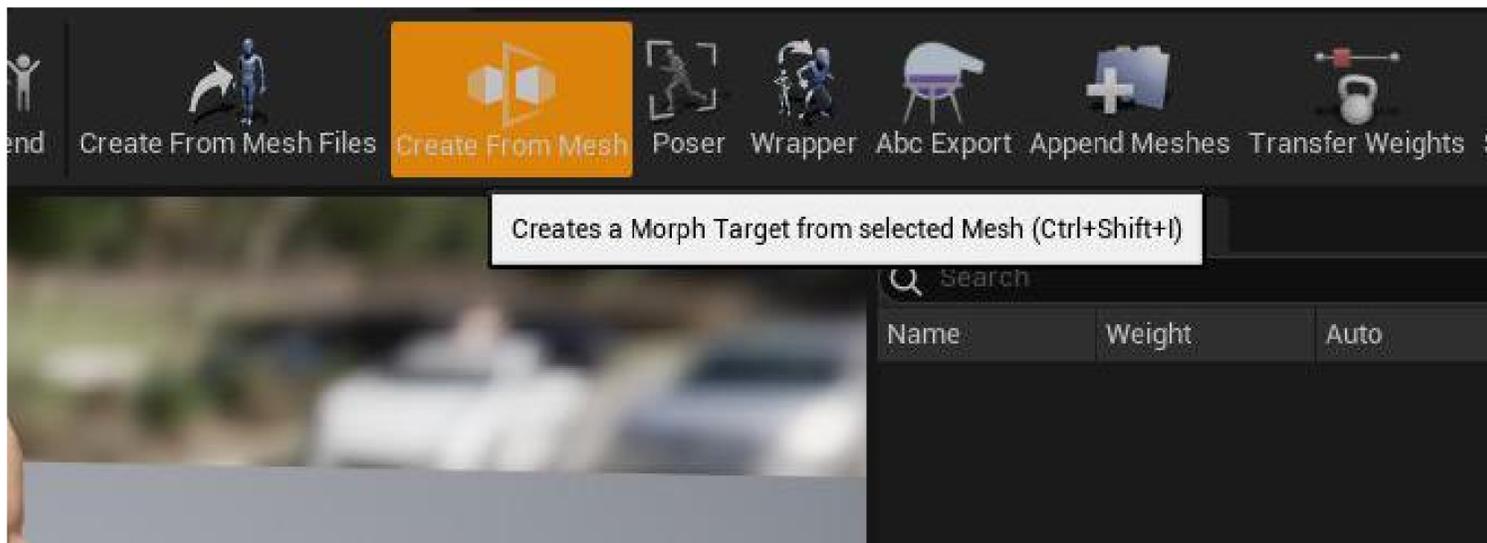


Figura 54. Morph Targets. Menú de Mesh Morpher para la creación de Morph Targets. Elaboración propia (2024).

Se importará el modelo base elaborado de MetaHuman y el modelo elaborado de *metahuman* transformado en el personaje en los campos respectivos. Esto servirá para trasladar el Blend Shape en formato de *Morph Target* a los propios *Skeletal Meshes* en Unreal Engine. Se asignará un nombre al *Morph Target* y, de esta manera, quedará listo.

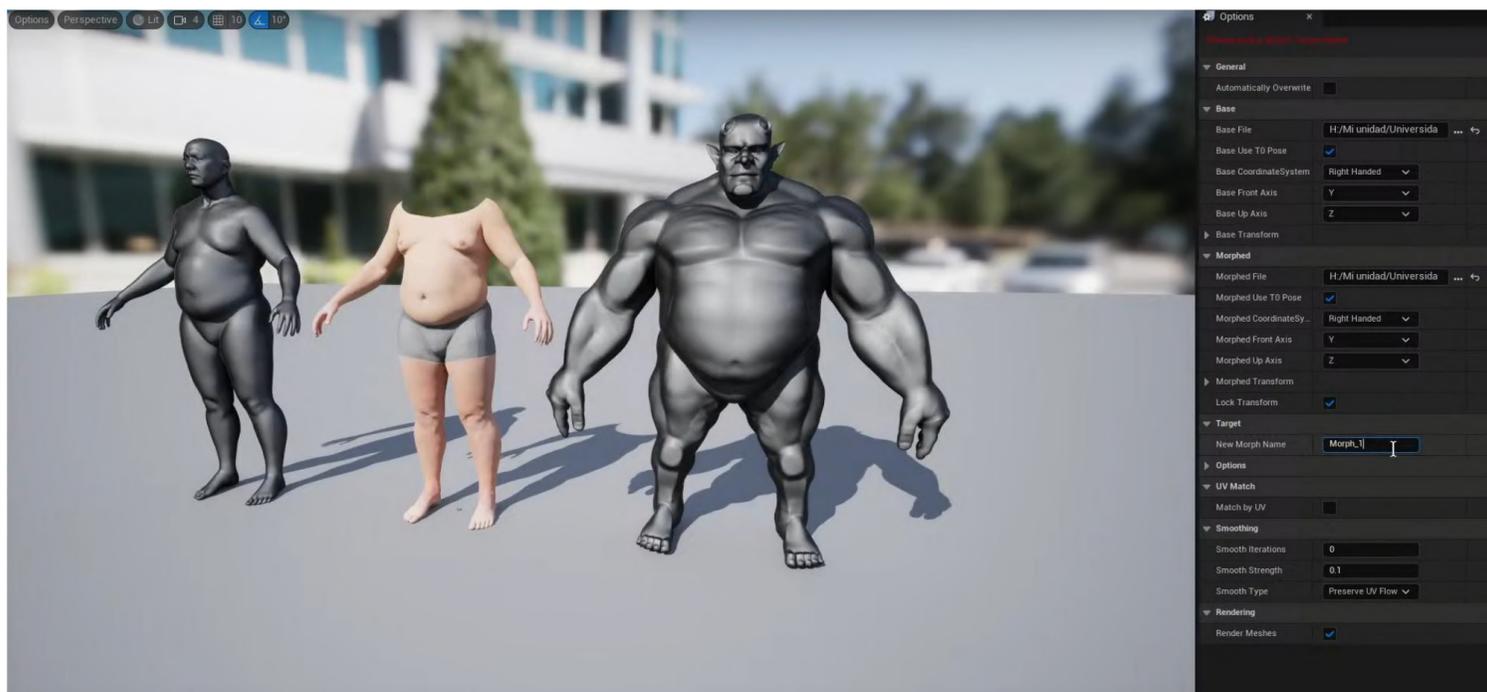


Figura 55. Morph Targets. Creación de Morph Targets de Noboru. Elaboración propia (2024).

Al encender los *Morph Targets* y colocar el valor en 1, se podrá observar cómo se modifica la forma del personaje. Este proceso debe realizarse tanto en el *Skeletal Mesh* del cuerpo como en el de la cabeza del *metahuman*.



Figura 55. Skeletal Mesh. Inspección del Skeletal Mesh de Noboru. Elaboración propia (2024).

En este punto, se podrá dirigir al material base del cuerpo y la cabeza del *metahuman* y realizar la importación de las nuevas texturas dentro del material ya creado para MetaHuman.



Figura 56. Material de Metahuman. Interfaz gráfica de material base del cuerpo. Elaboración propia (2024).

Al realizar el cambio en el material del *Face Mesh* es necesario también actualizar los mapas CM1, CM2, CM3, WM1, WM2 y WM3. En caso de únicamente actualizar el MAIN se producirán errores al ejecutar animaciones en el rostro.

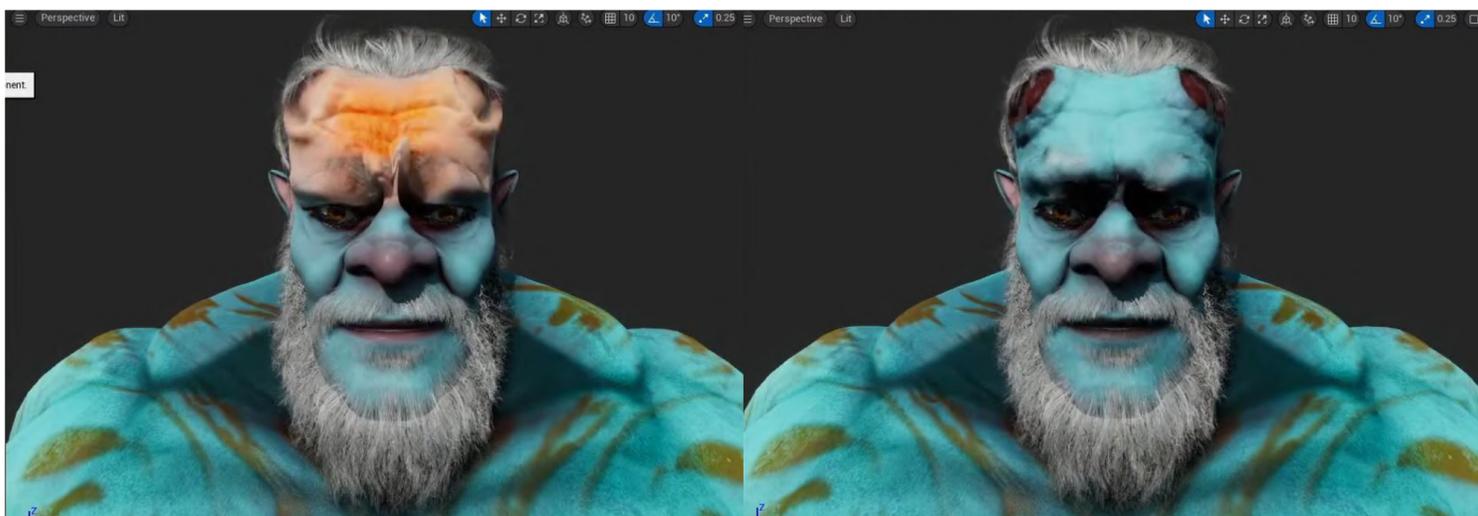


Figura 57. Material de Metahuman. Comparativa de normales de la cara de Noboru. Elaboración propia (2024).

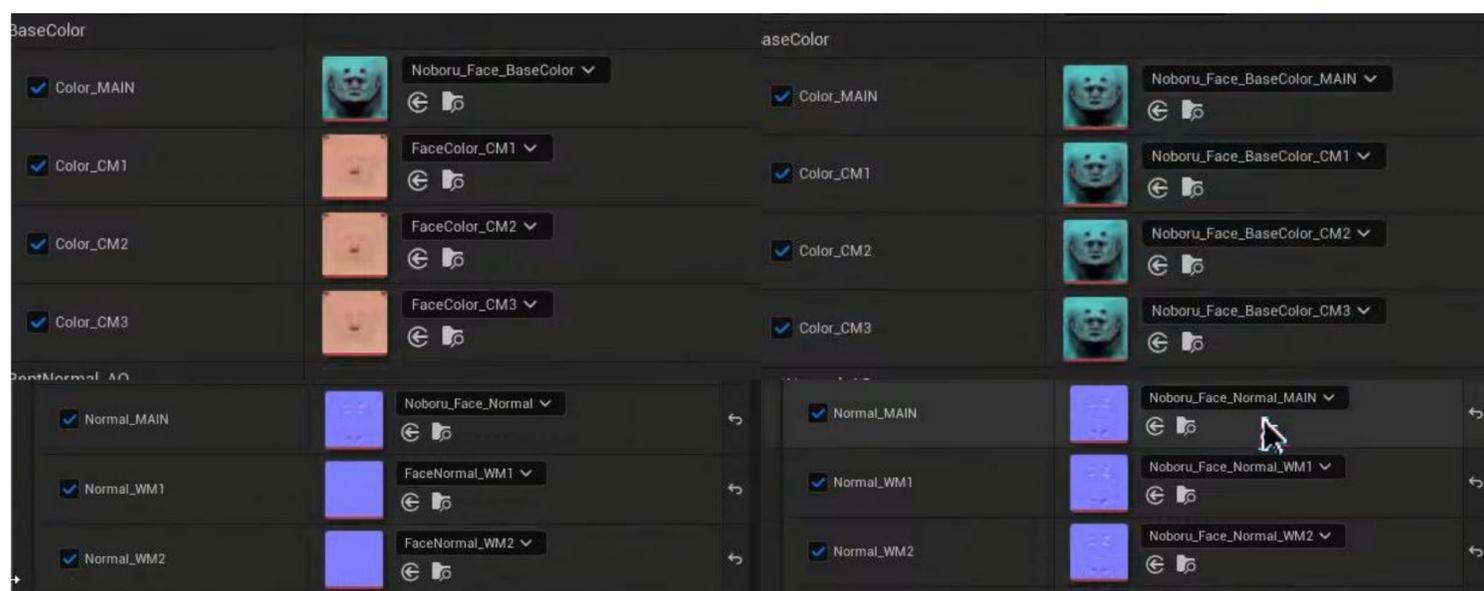


Figura 58. Material de Metahuman. Comparativa de normales de la cara de Noboru. Elaboración propia (2024).

En el caso de que el personaje tenga una estructura corporal más cercana al *metahuman* original, se podría omitir la etapa 2, pero debería ser reemplazada por un ajuste adicional:

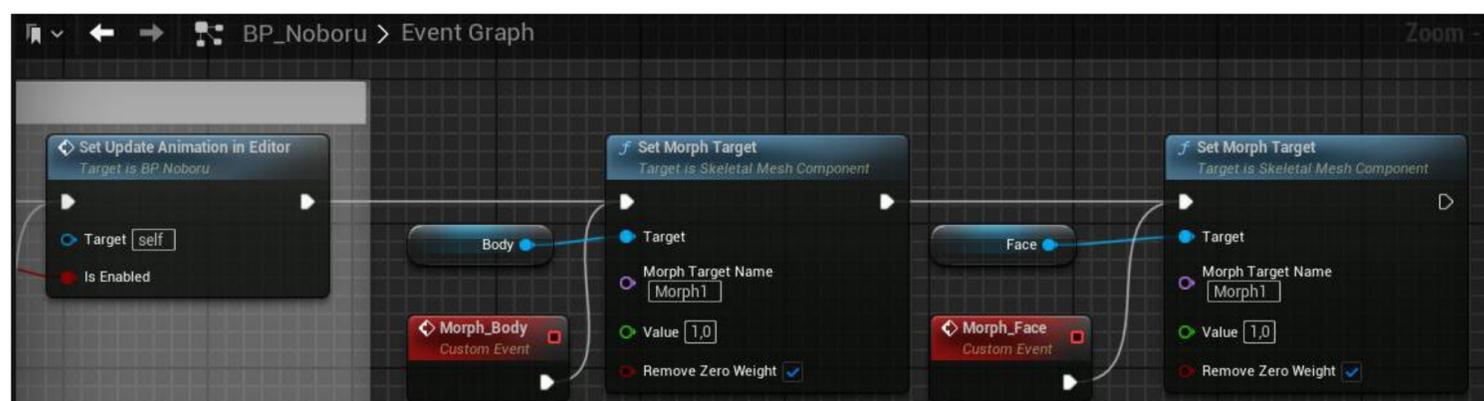


Figura 14. Event Graph. Interfaz gráfica del Event Graph de Metahuman. Elaboración propia (2024).

Como se observa en la imagen anterior, se debe añadir al *Event Graph* del *metahuman*, el nodo “*Set Morph Target*” e indicar el nombre del *MorphTarget* que se aplicará sobre el *Skeletal Mesh* y su valor en 1. Además, se deben crear los “*Custom Events*” llamados “*Morph_Body*” y “*Morph_Face*”, los cuales serán instanciados en el “*Construction Script*” del *metahuman*.

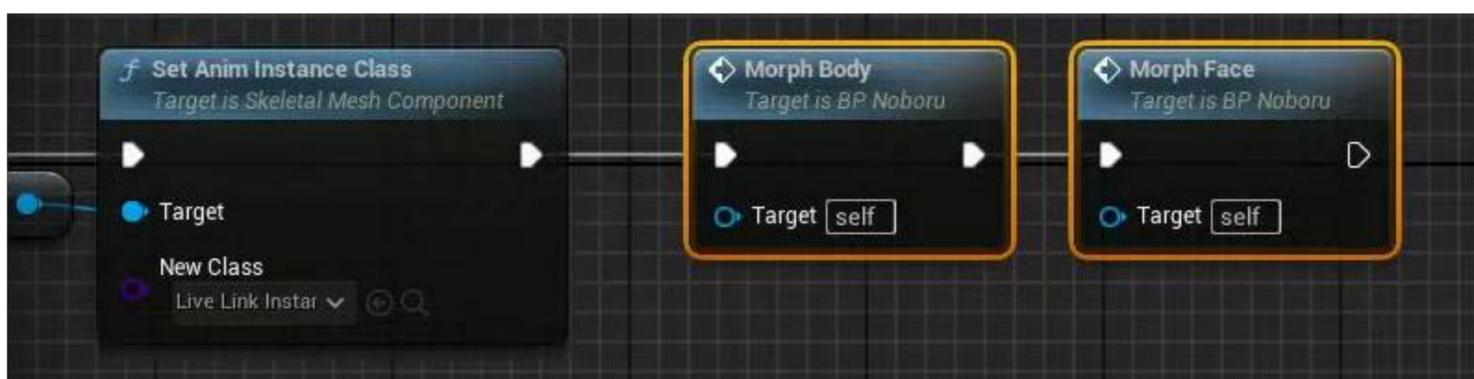


Figura 59. Construction Script. Interfaz gráfica del Construction Script de Metahuman. Elaboración propia (2024).

Asimismo, como paso adicional, se puede establecer en 0 el *Forced LOD* dentro del componente *LOD Sync* del *metahuman*, con el fin de preservar el máximo detalle en cinemáticas.



Figura 60. Level of Detail. Interfaz gráfica del Componente LOD Sync del personaje. Elaboración propia (2024).

En el componente *Skeleton*, “*Metahuman_base_skel*”, se puede añadir el *SK_Mannequin* en esqueletos compatibles y utilizar todas las animaciones predeterminadas dentro de *MetaHuman*.

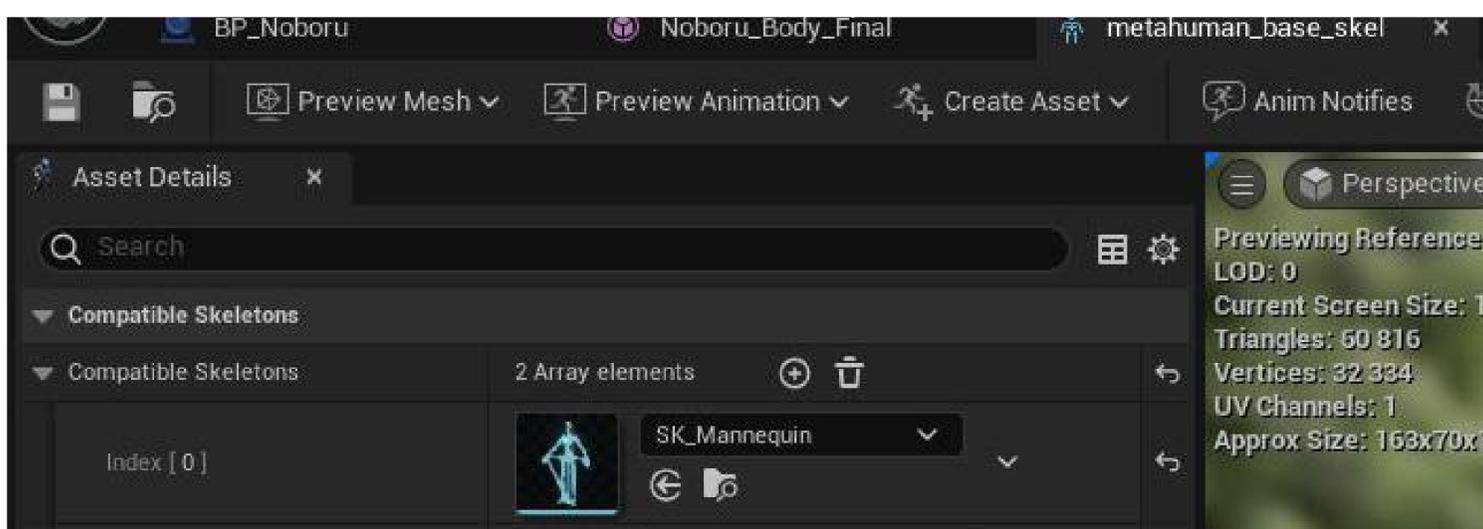


Figura 61. Skeleton de Metahuman. Compatibilidad de esqueletos en Metahuman. Elaboración propia (2024).

Etapa B: Ajuste de la estructura anatómica

Si se importa el propio *Blueprint* a un *Level Sequence*, en el caso del personaje elaborado, *Noboru*, se notará que los controladores y la jerarquía de los huesos del esqueleto no coinciden con la estructura anatómica del personaje debido a sus proporciones estilizadas.

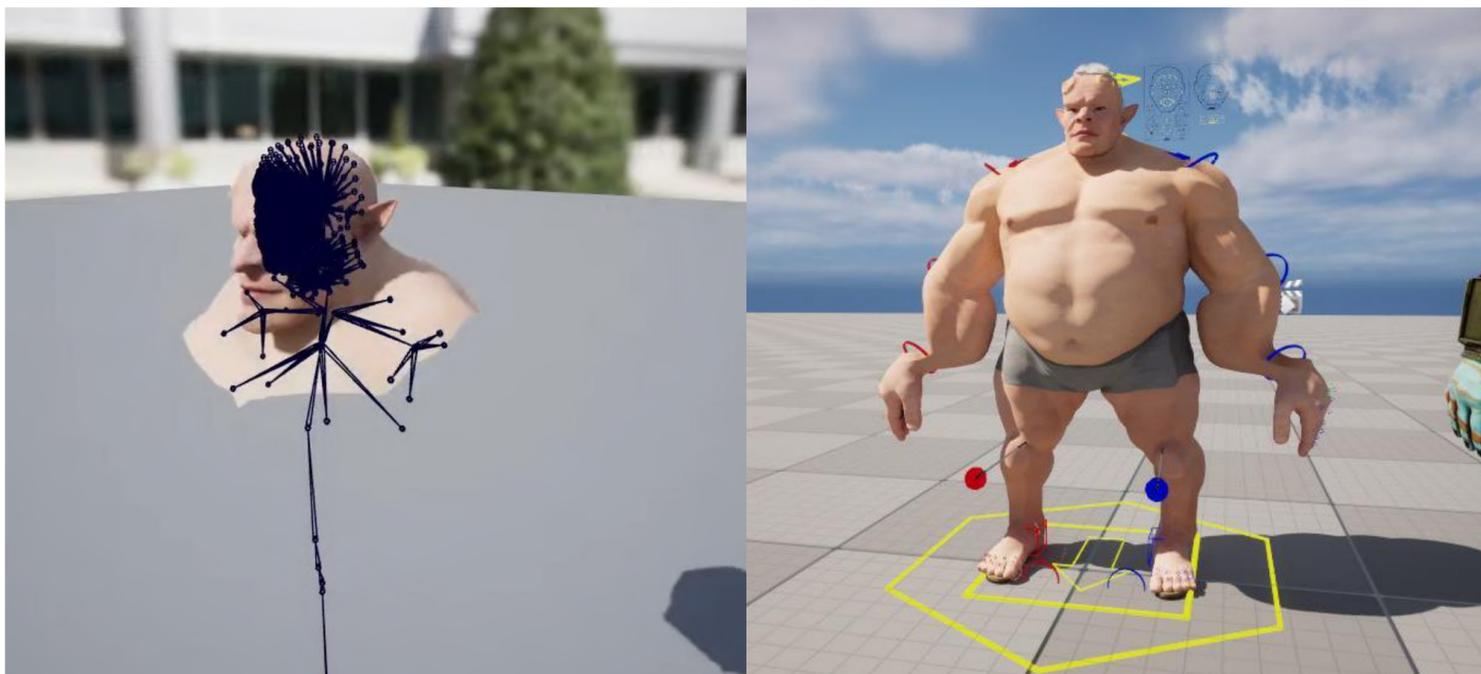


Figura 62. Errores del esqueleto. Problemas presentados sobre la estructura del esqueleto. Elaboración propia (2024).

Para solucionar esto, se volverán a abrir los propios *Skeletal Meshes* en *Mesh Morpher* y, esta vez, se utilizará la herramienta “*Poser*”. Este paso puede no ser necesario si el personaje tiene una estructura anatómica más realista o cercana a la del personaje original de MetaHuman, pero en este caso, era imprescindible.

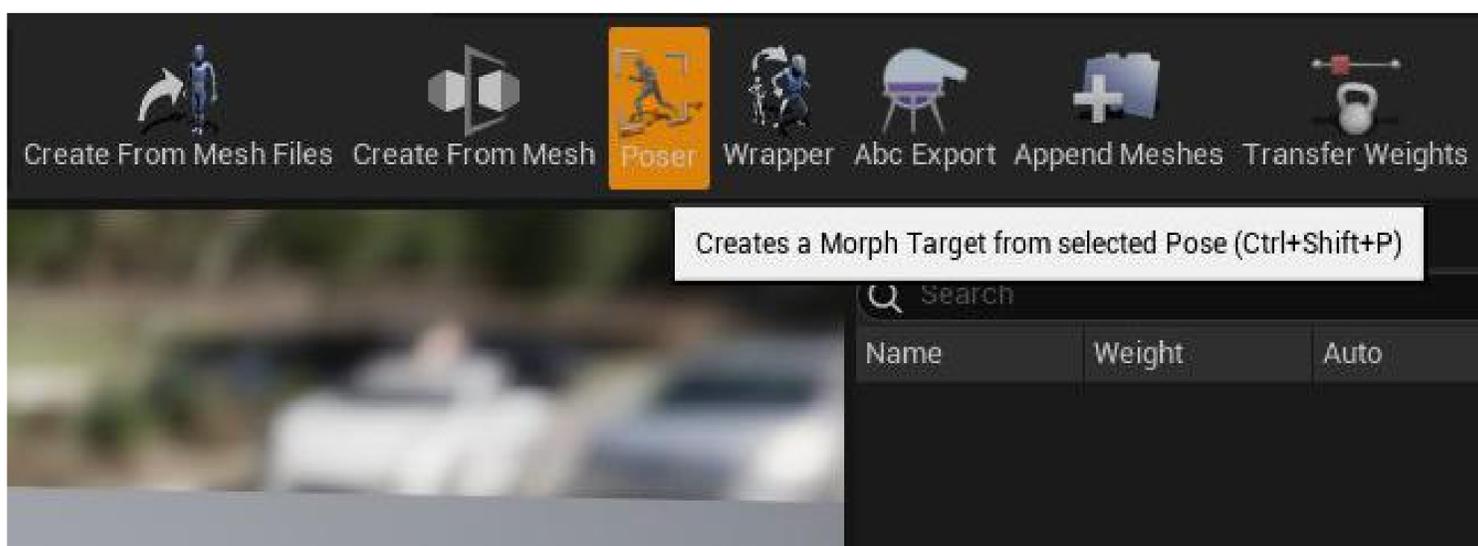


Figura 63. Interfaz de Mesh Morpher. Opción de “Poser” de Mesh Morpher. Elaboración propia (2024).

Al entrar al módulo de *Poser*, se seleccionará el propio *Morph Target* y se presionará el botón “*To bone snapshot*” y se guardarán los cambios. Es importante que el *Morph Target* lleve el mismo nombre tanto en el *Face Mesh* como en el *Body Mesh* del propio *metahuman*.

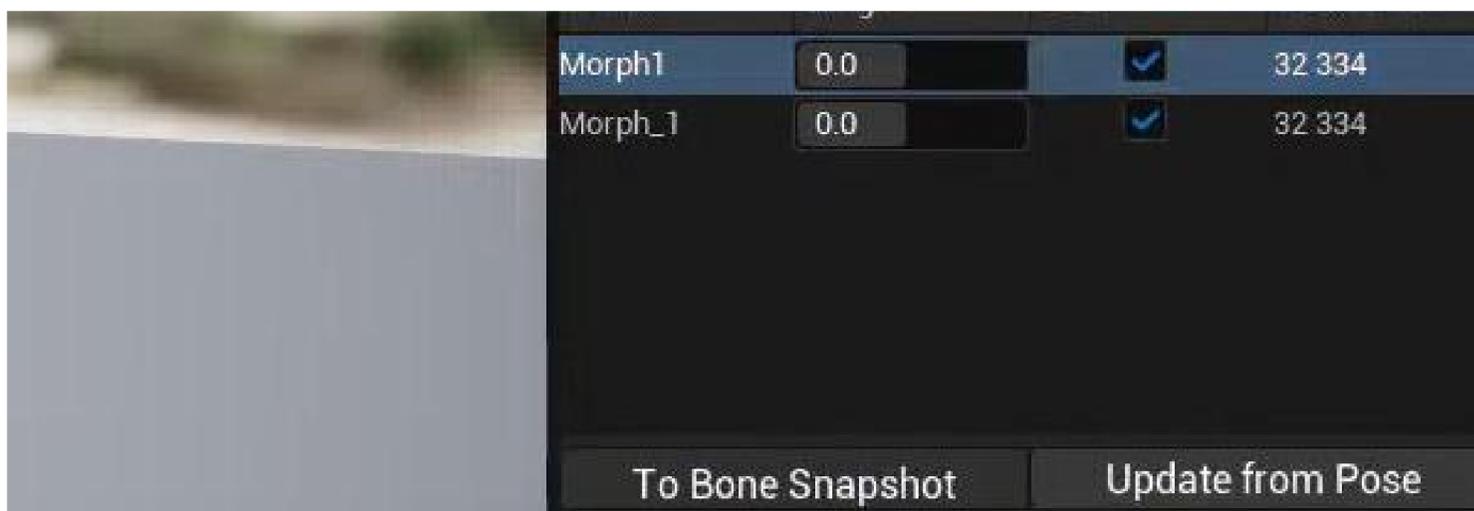


Figura 64. Interfaz de Mesh Morpher. Configuración de “Bone Snapshot” de Noboru. Elaboración propia (2024).

Al abrirse la ventana emergente para crear el “*Bone Snapshot*”, en el caso del *Face Mesh*, se debe digitar en “*Start Bone*” el hueso “*spine_05*”, mientras que en el *Body Mesh*, el “*Start Bone*” es el hueso “*root*”.

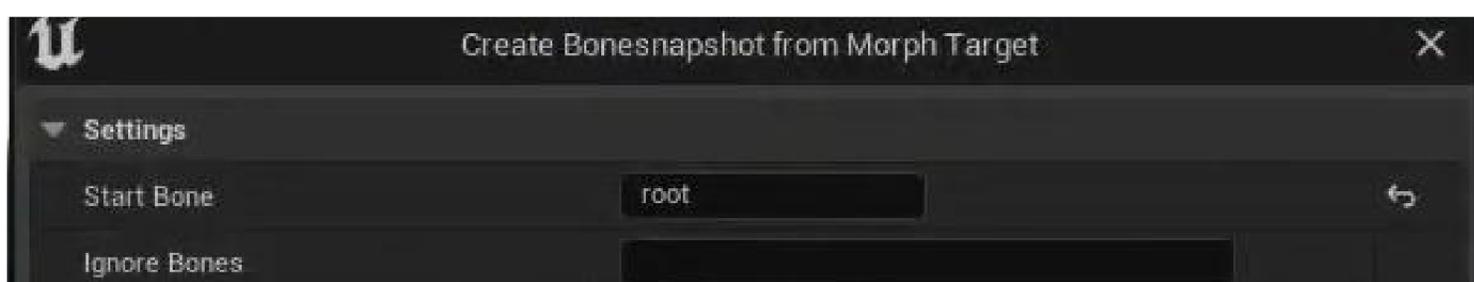


Figura 65. Interfaz de Mesh Morpher. Configuración de “Bone Snapshot” de Noboru. Elaboración propia (2024).

Posteriormente, dentro del módulo de “*Poser*”, se debe seleccionar la opción “*Edit - Merge Bonesnapshots*” e indicar el *snapshot* que se creó del cuerpo en el índice 0 y el *snapshot* de la cabeza en el índice 1.

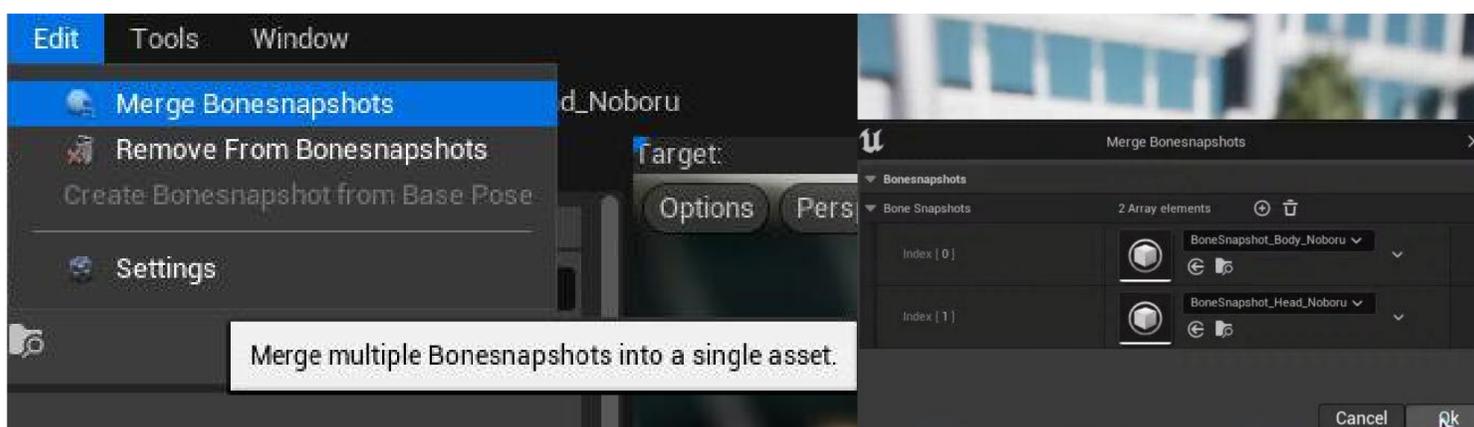


Figura 66. Interfaz de Mesh Morpher. Configuración de “Bone Snapshot” de Noboru. Elaboración propia (2024).

Una vez guardado este *snapshot* combinado, en el menú izquierdo, se seleccionará “*Bone Options*”, en el apartado “*Bone Snapshot*”. Posteriormente, en el apartado “*Morph Targets*”, se seleccionará el propio *Morph Target* y, posteriormente, se presionará el botón “*Update from pose*”.

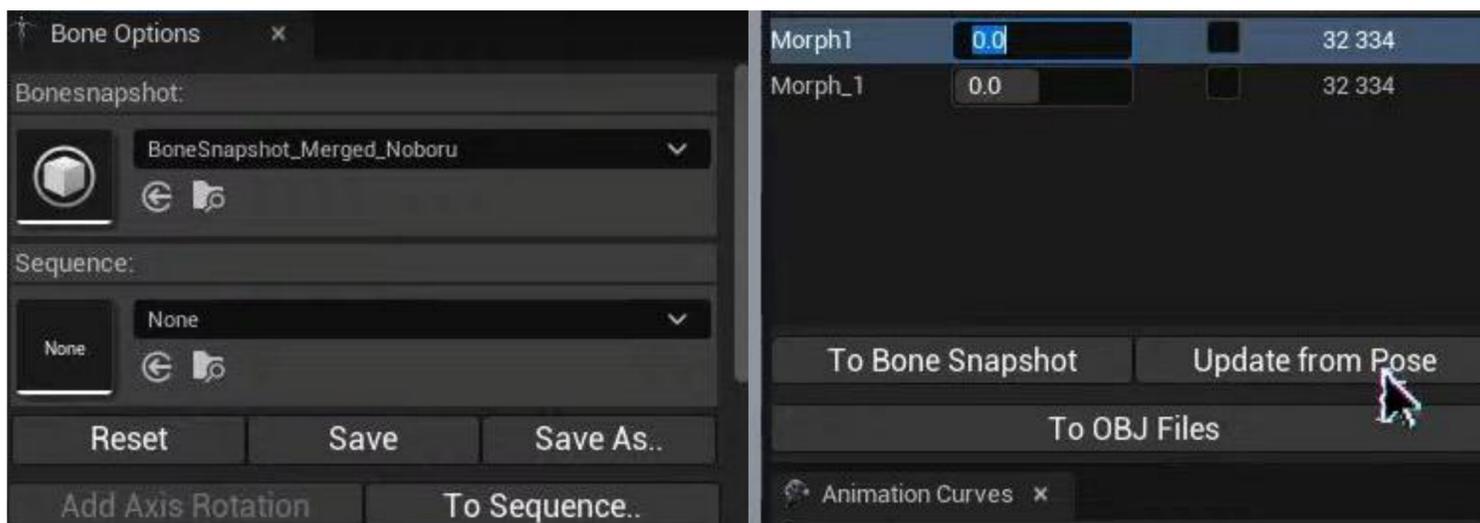


Figura 67. Interfaz de Mesh Morpher. Configuración de “Bone Snapshot” de Noboru. Elaboración propia (2024).

Esto ajustará el rig del personaje a la estructura anatómica del *Morph Target* que se creó. Finalmente, se encenderá el propio *Morph Target* y se presionará el botón “*Bake Skeleton*”. Este paso es irreversible, por lo que se debe verificar que todos los pasos se hayan ejecutado correctamente antes de hacer clic en “*Bake*”. Este proceso se realizará tanto en el cuerpo como en la cabeza.

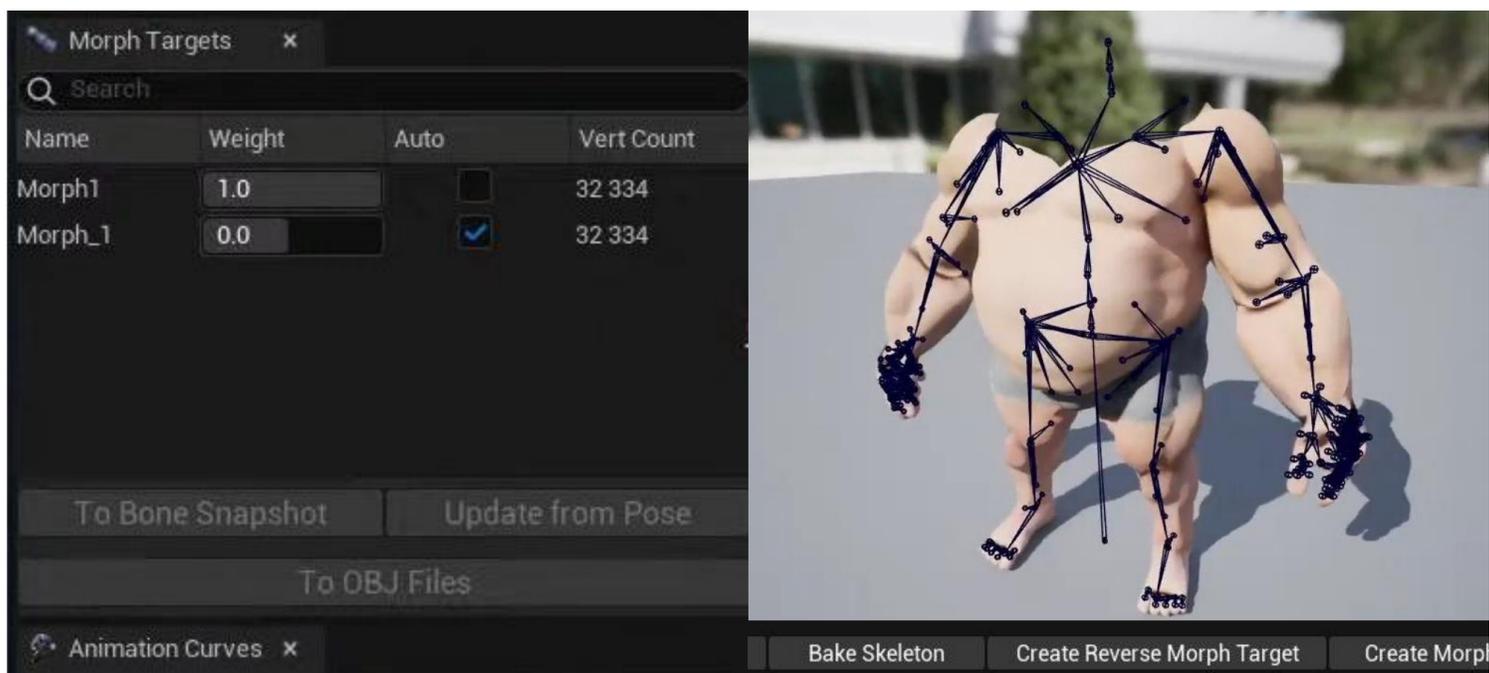


Figura 68. Interfaz de Mesh Morpher. Configuración de “Bone Snapshot” de Noboru. Elaboración propia (2024).

Finalmente, se deberá aplicar el proceso de *Recompute the Normals* para evitar cualquier conflicto con la malla.

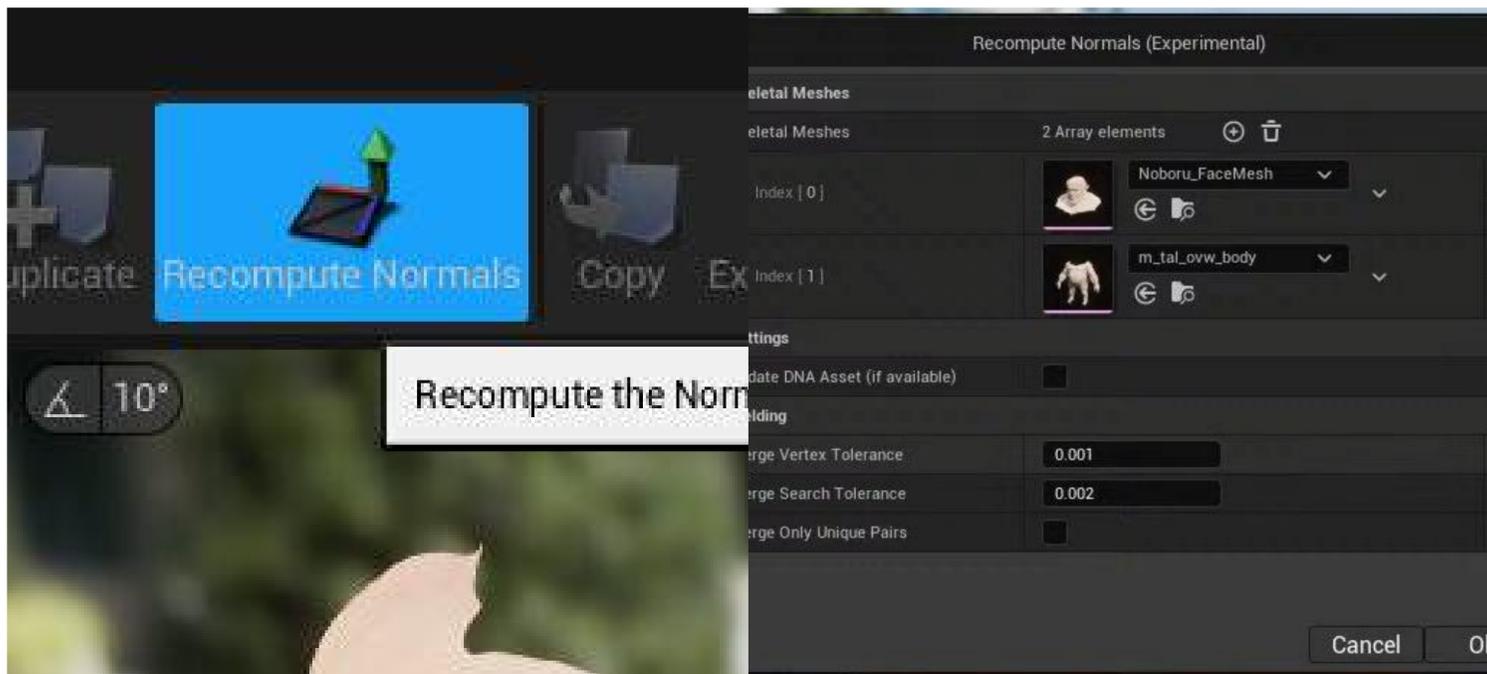


Figura 69. Interfaz de Mesh Morpher. Opción de “Recompute Normals” del personaje. Elaboración propia (2024).

En el presente caso, surgió un único problema con la estructura del *rig*, donde la estructura de *Neck* y *Head* estaban desfasadas con respecto a la nuca del personaje. Si esto sucede con cualquiera de los huesos de tu personaje, la solución más eficiente es utilizar la herramienta de *Edit Skeleton* del plugin de *Editing Tools* para posicionar los *joints* de forma correcta. No es recomendable hacer esta edición y reimportación con algún software como Maya o Blender, ya que esto casi seguramente generará problemas de compatibilidad y se podrían perder *Morph Targets*, pintados de pesos o funcionalidades del *metahuman*.

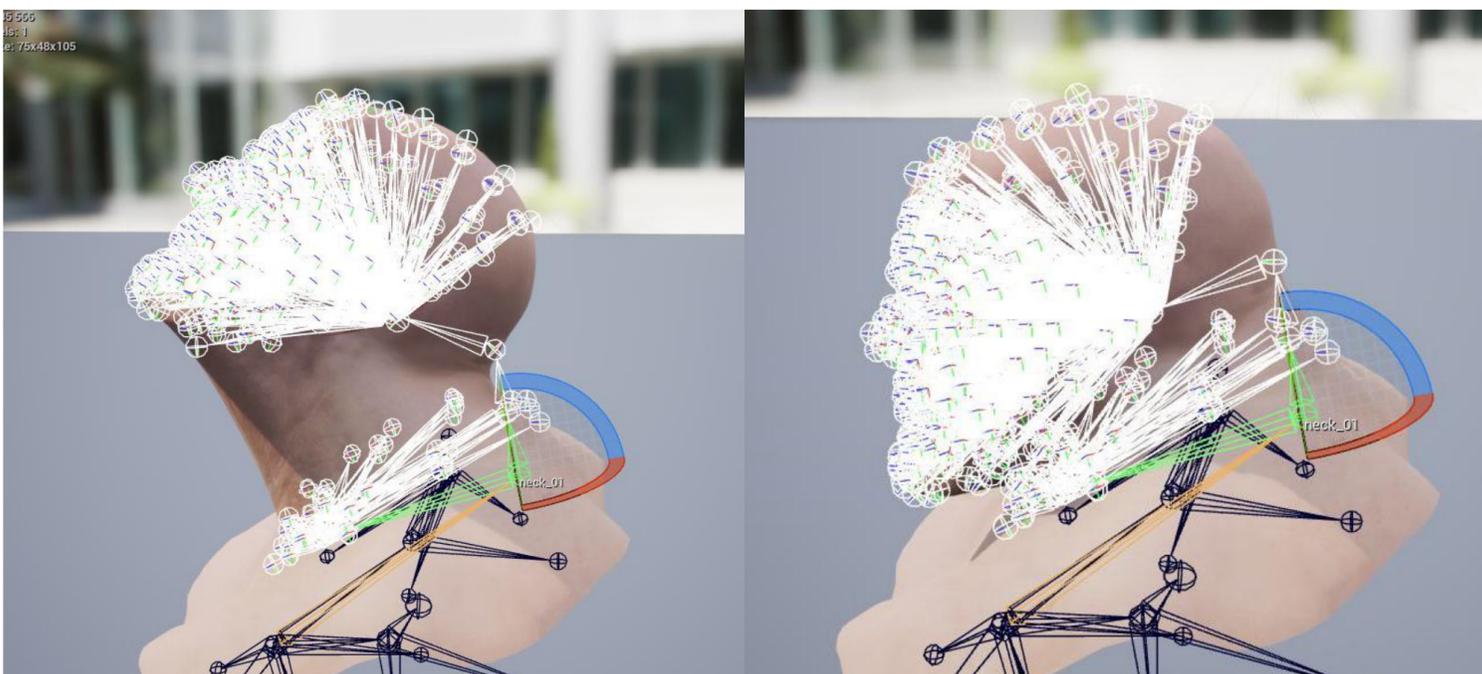


Figura 70. Errores en el rig. Problemas ocasionados tras Bone Snapshot. Elaboración propia (2024).

Una vez que se selecciona la opción “*Edit Skeleton*”, se deberá ajustar el posicionamiento de los huesos con las herramientas de traslación y rotación. Seguidamente, se podrá presionar el botón “*Accept*”.

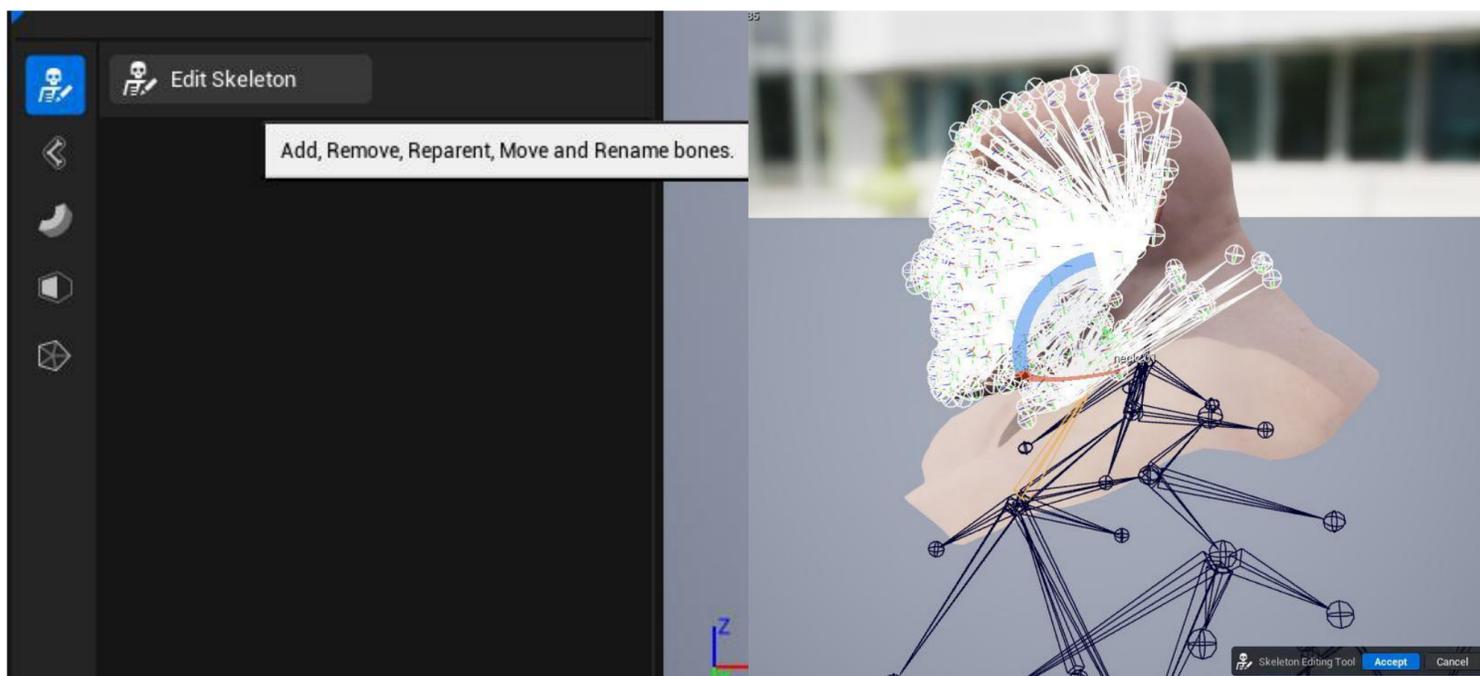


Figura 71. Editing Tools. Corrección de problemas ocasionados tras Bone Snapshot. Elaboración propia (2024).

Etapa C: Ajuste del control rig

Aunque el rig corporal fue actualizado, sus controladores se mantuvieron en su posición original. La solución indicada ante esta problemática es recrear el Control Rig corporal con las funcionalidades recientemente implementadas en Unreal Engine 5.4 sobre la creación de *rigs* modulares.

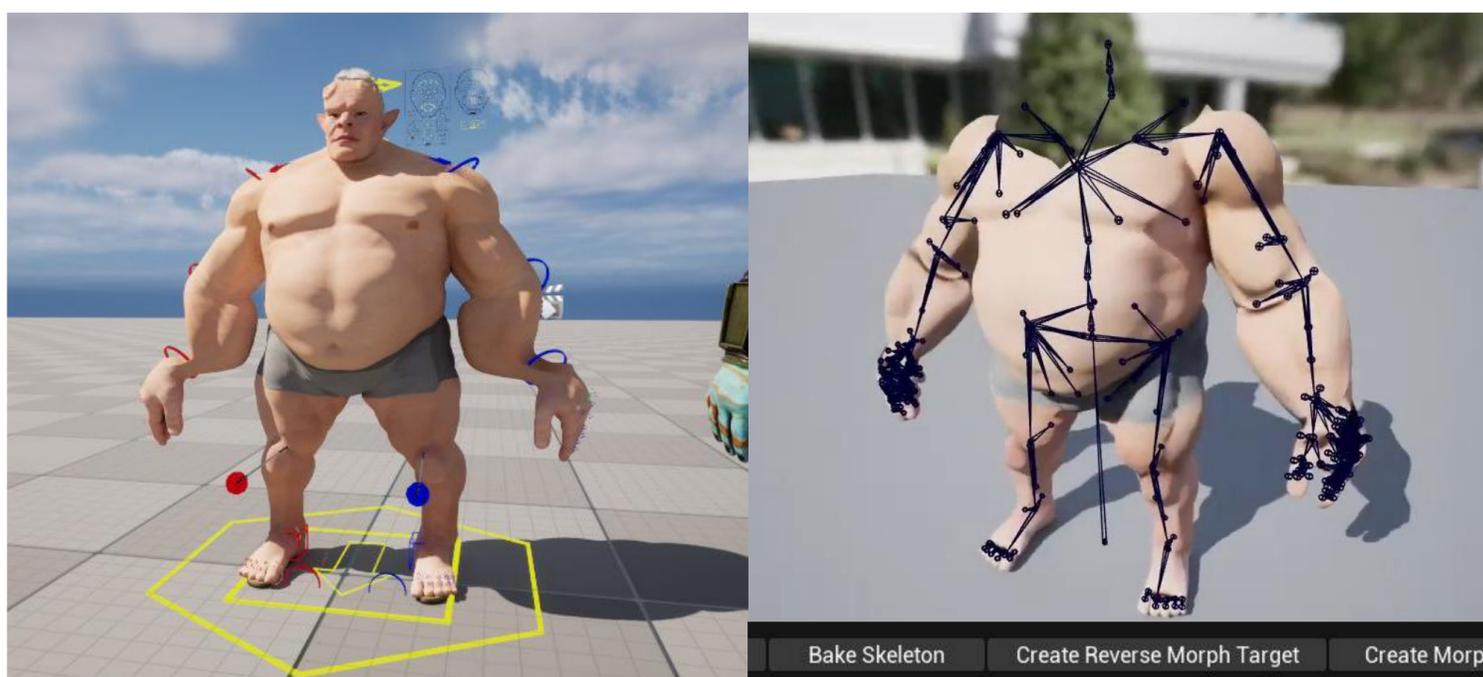


Figura 72. Editing Tools. Corrección de problemas ocasionados tras Bone Snapshot. Elaboración propia (2024).

Para esto, se dará clic derecho sobre el *Blueprint* de MetaHuman y se presionará “*Animation - Control Rig - Modular Control Rig*”. Esta herramienta es compatible con el rig de MetaHuman y garantiza ajustar el Control Rig en un lapso muy corto y adaptado a la estructura del propio *metahuman*. Únicamente se deben arrastrar los componentes necesarios a la estructura del rig y, de ser necesario, ajustar el tamaño y color de los controladores en los detalles del componente. Además, es necesario actualizar el nuevo Control Rig dentro del *Skeletal Mesh* del cuerpo.

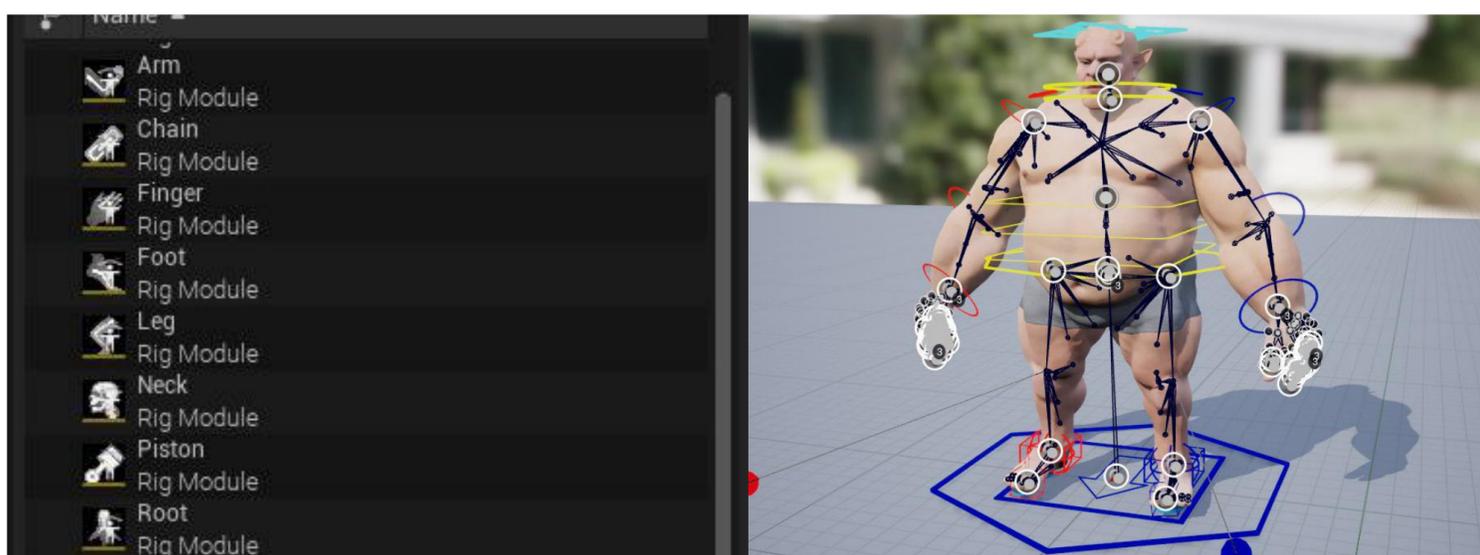


Figura 73. Modular Control Rig. Creación de rig modular en Nuboru. Elaboración propia (2024).

Etapa D: Adición de la ropa

Para añadir la ropa al *metahuman*, al igual que con el propio rig de Advanced Skeleton, se realizará la exportación de los propios *Skeletal Meshes*. En este caso, se utilizará Blender y se exportarán los objetos que conformarán cada *Skeletal Mesh* aunados al rig de MetaHuman, recordando haber completado el pintado de pesos de los objetos. Se debe mantener la selección de “*Selected objects*”, “*Armature*” y “*Mesh*”; también es importante desmarcar la opción de “*Add leaf bones*”.

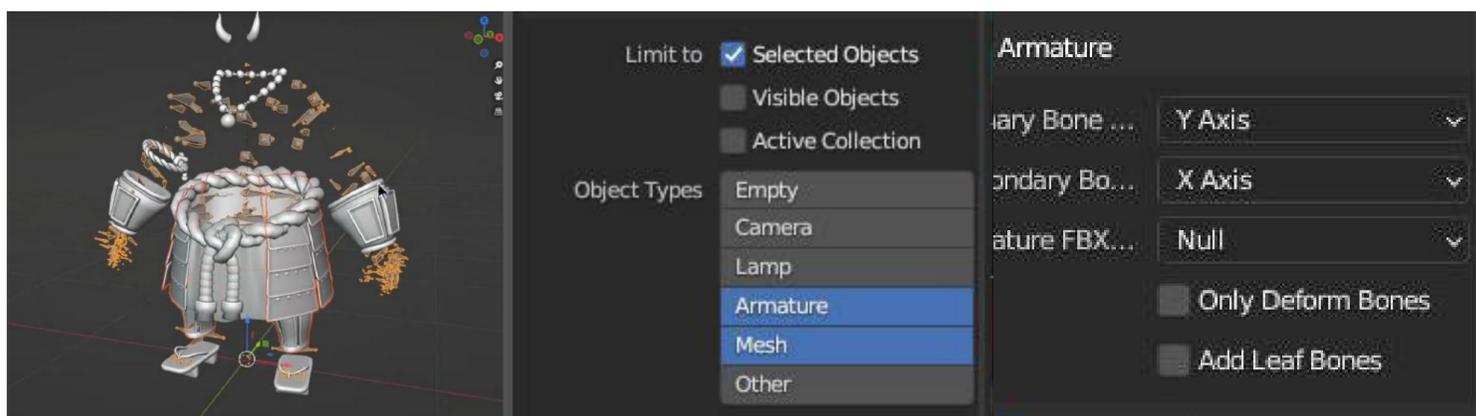


Figura 74. Ropa de Nuboru. Interfaz gráfica de exportación FBX. Elaboración propia (2024).

Al importar estos *Skeletal Meshes* a Unreal Engine, se realizará la configuración del material y se asignará correctamente a cada uno de los *Skeletal Meshes*. Asimismo, se eliminarán todos los *Physics Assets* y se reutilizará el original del cuerpo del *metahuman* en cada una de las prendas de ropa. Esto se analizará con mayor profundidad en el capítulo 5.

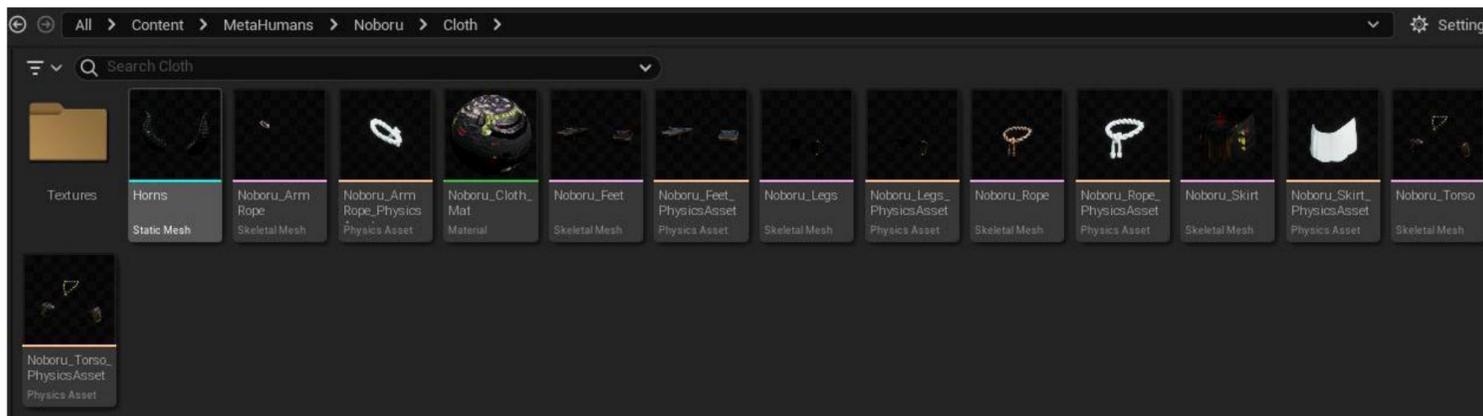


Figura 75. Content Browser. Importación de assets de Noboru en Unreal Engine. Elaboración propia (2024).

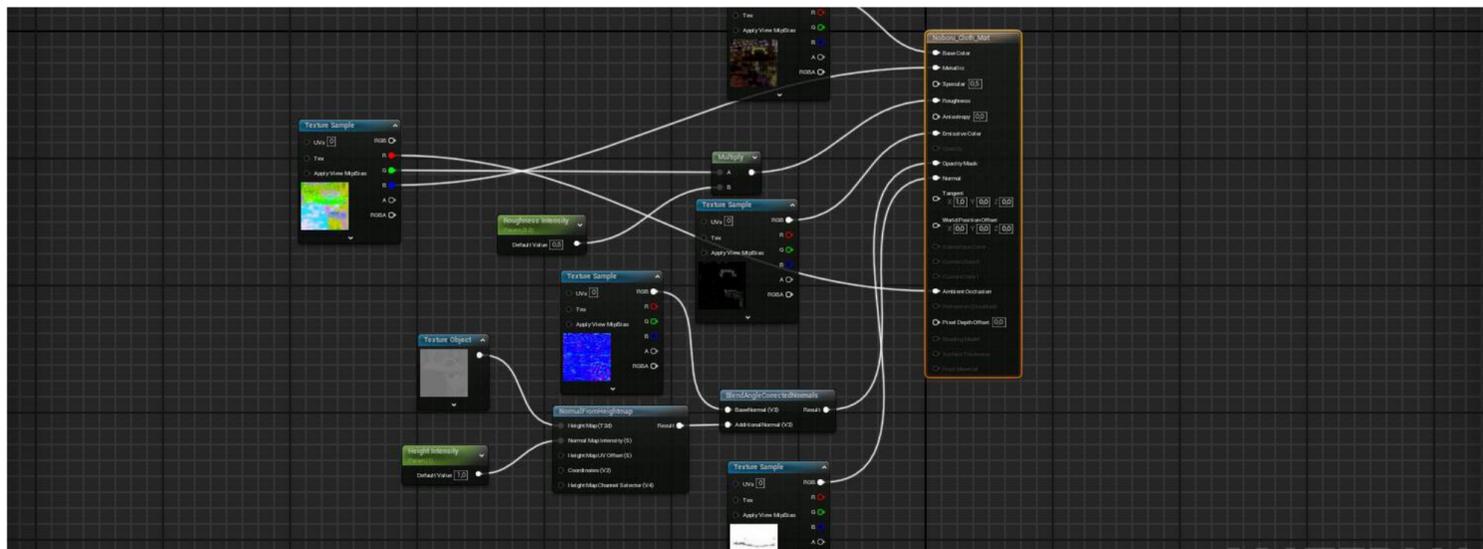


Figura 76. Material de la ropa. Nodos del material de la ropa de Noboru. Elaboración propia (2024).

Estos *Skeletal Meshes* ya pueden ser asignados en la jerarquía del propio *Blueprint* e incluso se pueden añadir *Skeletal Meshes* adicionales a los proporcionados por *Metahuman*, al actualizar el propio *construction script*.



Figura 77. Construction Script de Noboru. Nodos y jerarquía del blueprint de Metahuman. Elaboración propia (2024).

Etapa E: Configuración necesaria para animación

Para el apartado de animación, MetaHuman requiere un último paso, en el cual se editará el componente *Face_Archetype_Skeleton* y se añadirá un *Blend Mask* en la jerarquía de los huesos. En este sentido, se establecerá el valor “0.5” en el hueso “Head”.

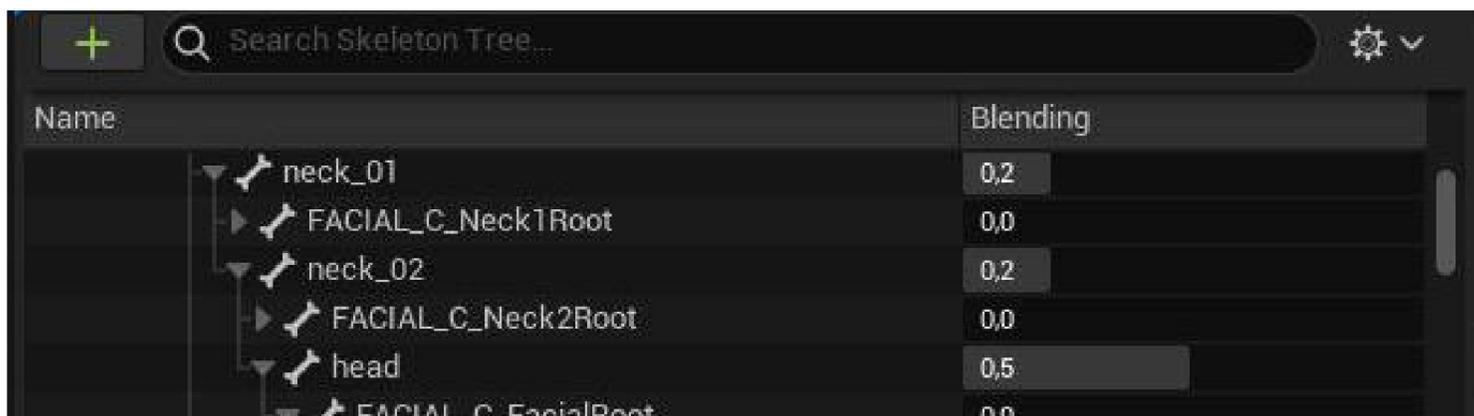


Figura 78. Jerarquía de Esqueleto. Configuración del asset *Face_Archetype_Skeleton*. Elaboración propia (2024).

Posteriormente, se editará el componente *Face_AnimBP* y se establecerá en las propiedades del nodo “*Layered blend per bone*”, el *Blend Mode* en *Blend Mask*.

Además, se añadirá el *Blend Mask* que se creó en la opción “*Blend Masks*” y, en el menú “*Curve Blend Option*”, se seleccionará “*Normalize by Weight*”.

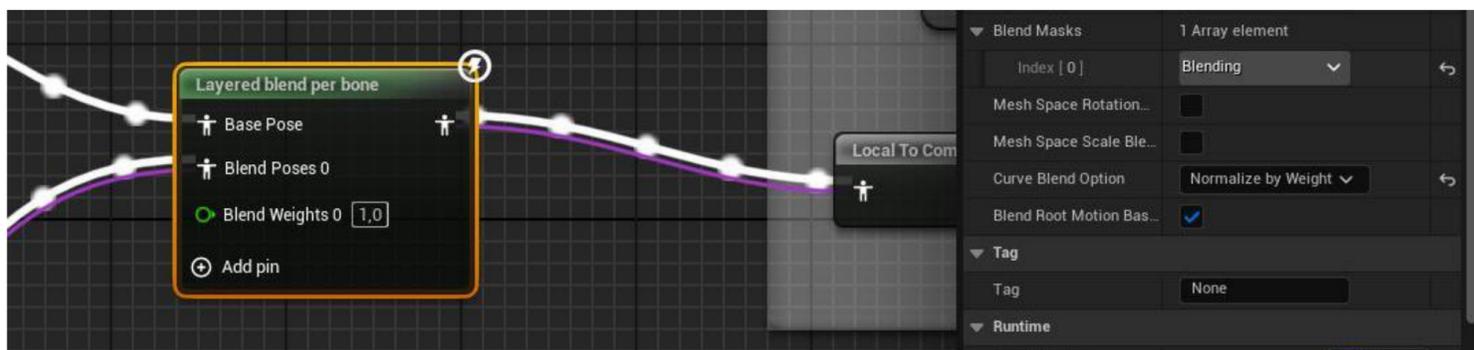


Figura 79. Jerarquía de Esqueleto. Configuración del asset *Face_Archetype_Skeleton*. Elaboración propia (2024).

En este punto, ya se tendrá el *metahuman* completamente listo y funcional dentro de *Unreal Engine*. Solo queda pendiente repasar la configuración de simulaciones y dinámicas, al igual que con el rig de *Advanced Skeleton*. En el caso de MetaHuman, ya puede asignarse cualquier *Animation Sequence* compatible con el cuerpo y/o rostro del personaje.

CAPÍTULO 5

SIMULACIONES Y FÍSICAS DE UNREAL ENGINE

1. Simulaciones de pelo y tela

Este apartado es bastante complejo; por lo tanto, se desglosará en varias secciones para facilitar la comprensión.

Sección 1: Simulaciones de pelo

Para realizar las simulaciones de pelo y crear los *Groom assets* del personaje elaborado en este proyecto, se utilizaron las funciones del nuevo sistema de *Grooming* de Blender. Se consideró que esta herramienta es accesible y eficiente para esculpir el pelo, aunque también se podría haber utilizado *XGen* en Autodesk Maya u otras similares.

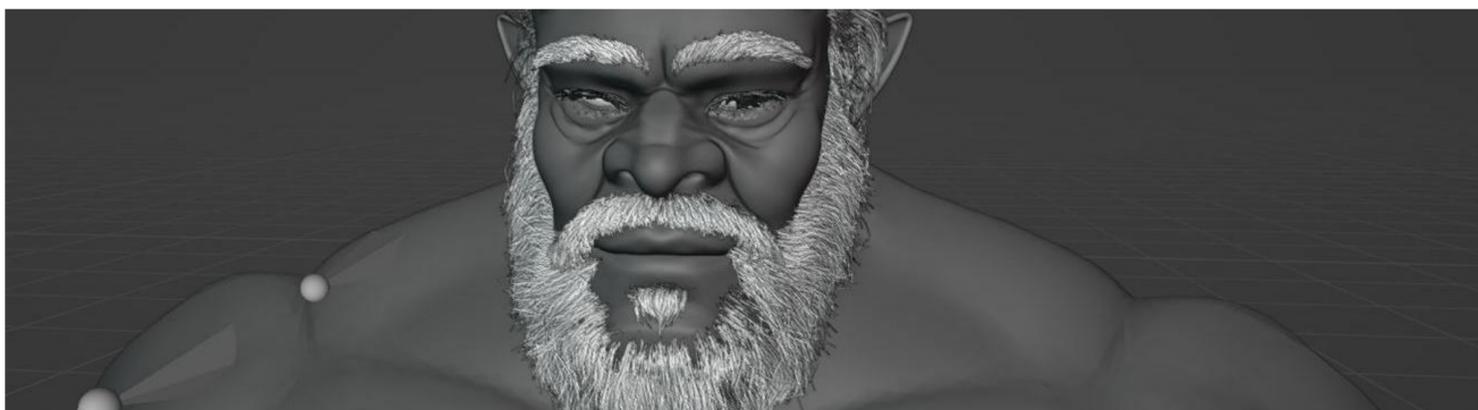


Figura 80. Grooming de Noboru. Diseño del grooming del personaje. Elaboración propia (2024).

Preparación del cabello y vello facial en Blender:

- Una vez el cabello y el vello facial estén listos, deben ser emparentados al *Mesh* que contiene el *grooming* dentro de Blender.
- Para exportar a Unreal Engine, utilicé el *plugin Groom Exporter* de Blender. [<https://turbocheke.gumroad.com/l/Groomexporter>]

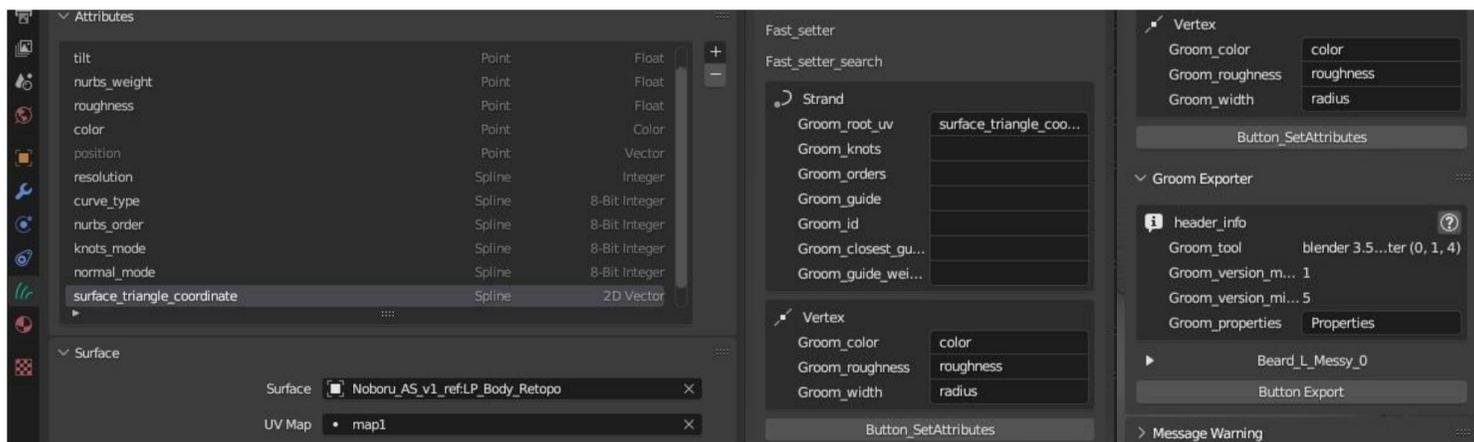


Figura 81. Exportación de Groom Assets. Configuración del Grooming dentro de Blender y plugin de Groom Exporter. Elaboración propia (2024).

Configuración en Unreal Engine:

- Activar los *plugins Alembic Importer* y *Alembic Groom Importer* en Unreal Engine.
- Exportar los componentes de *grooming* desde Blender como *Alembic* utilizando el *Groom Exporter*.
- Importar estos componentes arrastrándolos hacia el *Content Browser* de Unreal Engine.



Figura 82. Importación de Groom Assets. Menú de importación de Groom Assets de Noboru. Elaboración propia (2024).

Ajustes y vinculación en Unreal Engine:

- Ajustar la escala y activar o desactivar las físicas sobre cada componente de grooming según sea necesario.

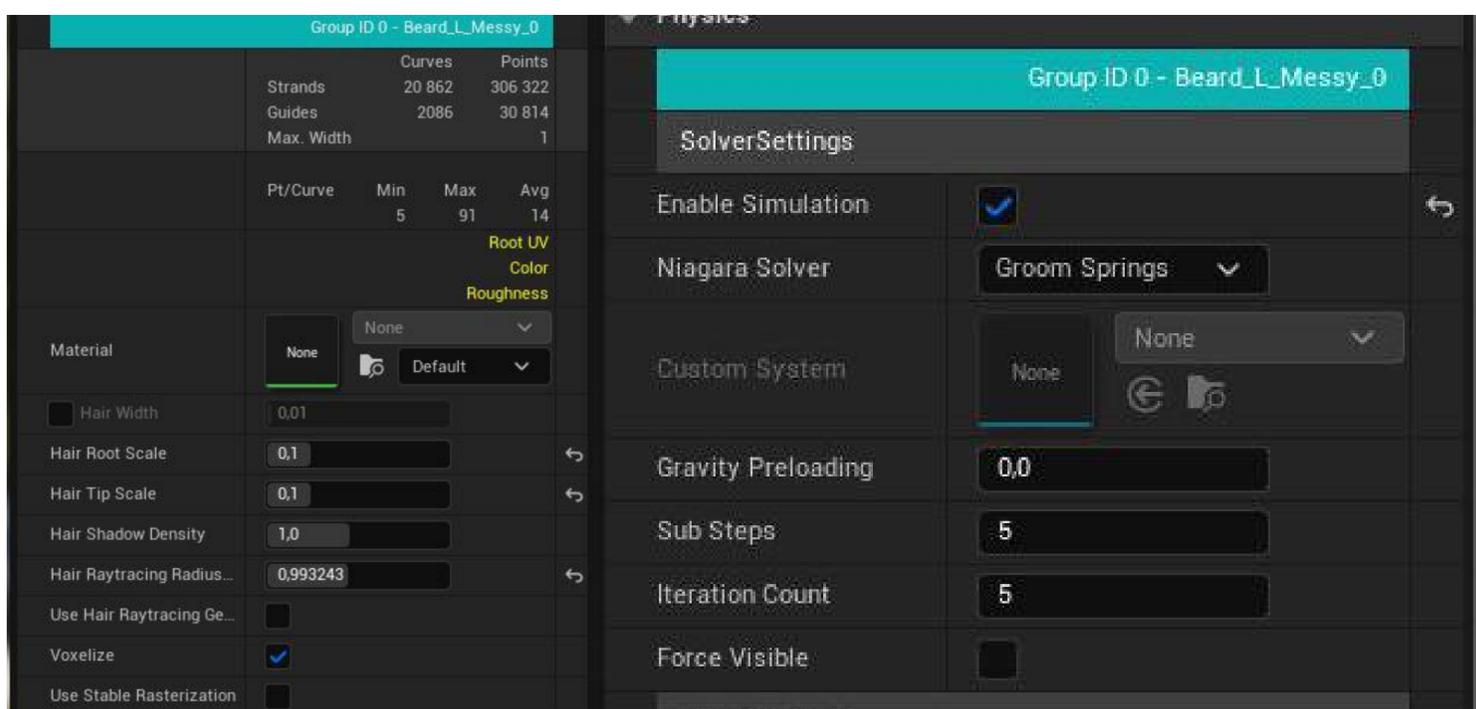


Figura 83. Groom Assets. Configuración de simulaciones de Grooming. Elaboración propia (2024).

- Asegurarse de que los *Groom assets* sean componentes hijo dentro del *Blueprint* del *Skeletal Mesh* al que fueron emparentados durante su creación y exportación en *Blender*.
 - Para *MetaHuman*, emparentar al *Face Mesh*.
 - Para *Advanced Skeleton*, emparentar al *Body*.

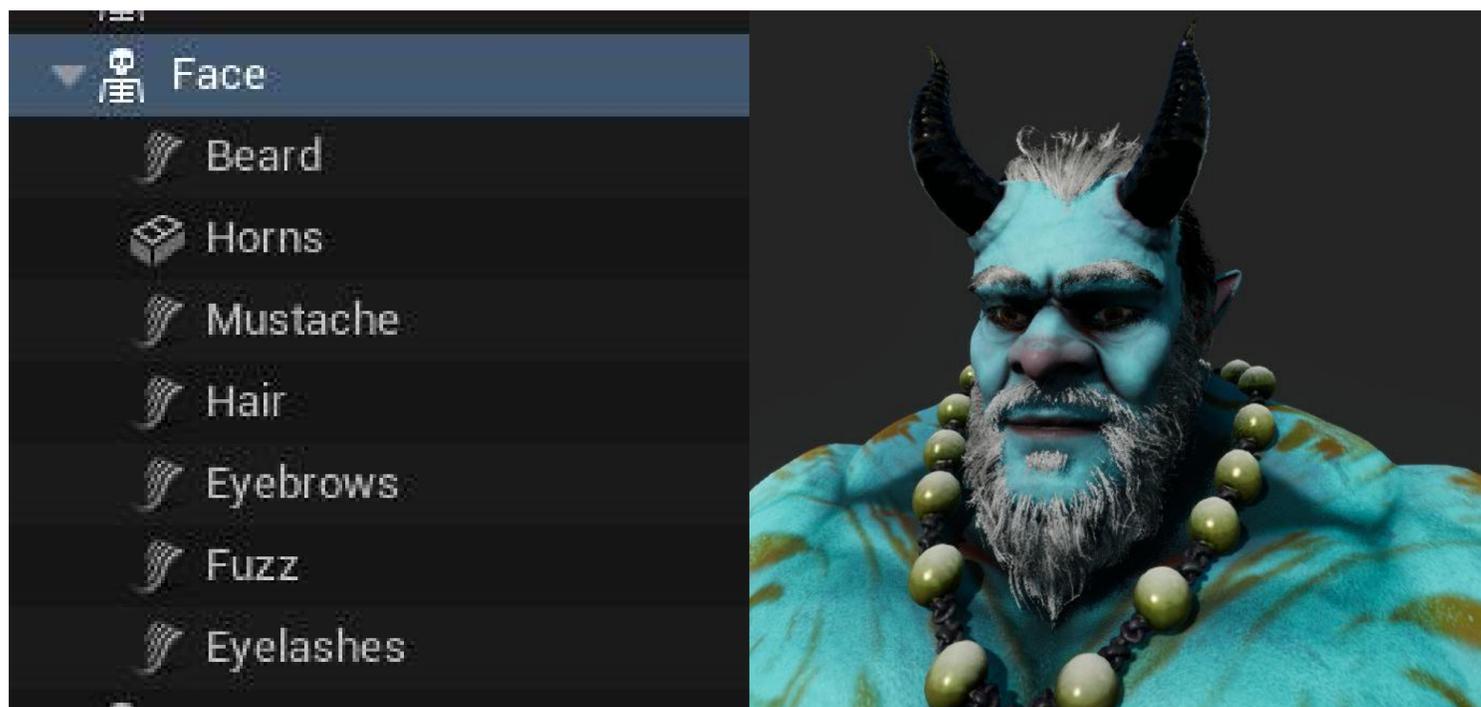


Figura 84. Groom Assets. Jerarquía de Groom Assets de Metahuman. Elaboración propia (2024).

- En el Content Browser, seleccionar nuestro "*Groom asset*" y presionar, "Clic derecho - *Create - Binding Asset*". Se abrirá un menú, en el que se debe especificar en "*Target Skeletal Mesh*" el *Skeletal Mesh* correspondiente, ya sea, el de *MetaHuman*, o el de *Advanced Skeleton*, según sea el *Blueprint* que se está trabajando. Este *Binding Asset* resultante debe ser asignado en los detalles del componente dentro del *Blueprint* del personaje.

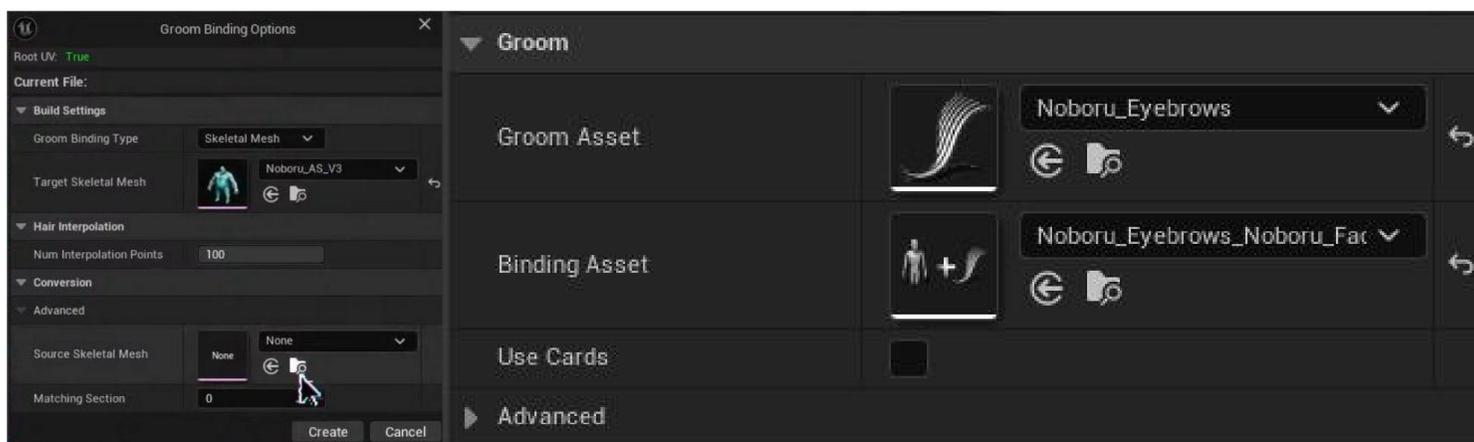


Figura 85. Físicas de Noboru. Binding Assets del Grooming de Noboru. Elaboración propia (2024).

Una vez finalizado, se debería poder observar todos los *Groom assets* correctamente asignados y conectados al cuerpo del personaje.

Sección 2: Simulaciones de tela

La siguiente etapa es aplicar simulaciones de tela a las prendas de ropa. Los pasos son los siguientes:

Creación de *Clothing Data*:

- Abrir el *Skeletal Mesh* de las prendas de ropa a las que se desea aplicar una simulación de tela.
- Seleccionar el *mesh*, hacer clic derecho y seleccionar la opción “*Create Clothing Data from Selection*”.

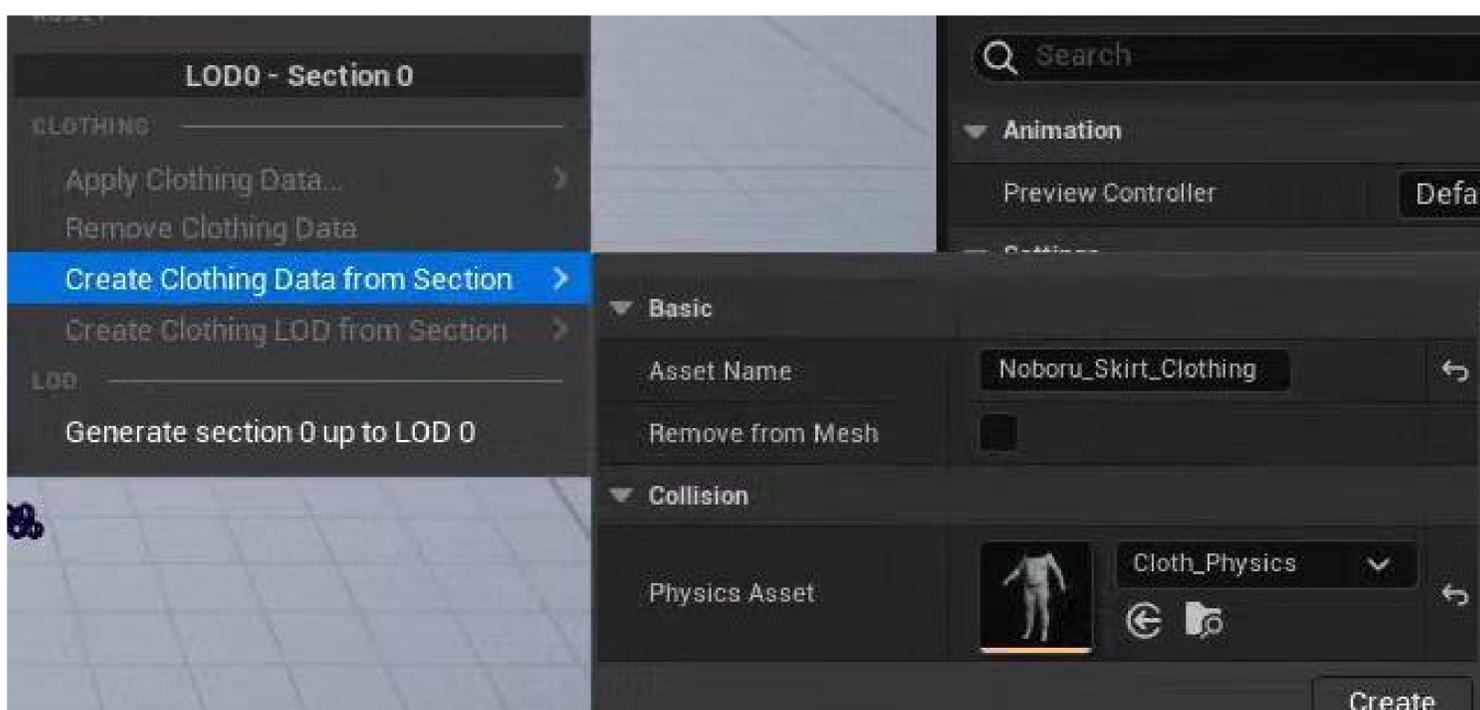


Figura 86. Físicas de Noboru. Cloth Paint de Noboru. Elaboración propia (2024).

- Seleccionar el *Clothing Data* en el tab “*Clothing*”, hacer clic derecho y seleccionar “*Apply Clothing Data*”.

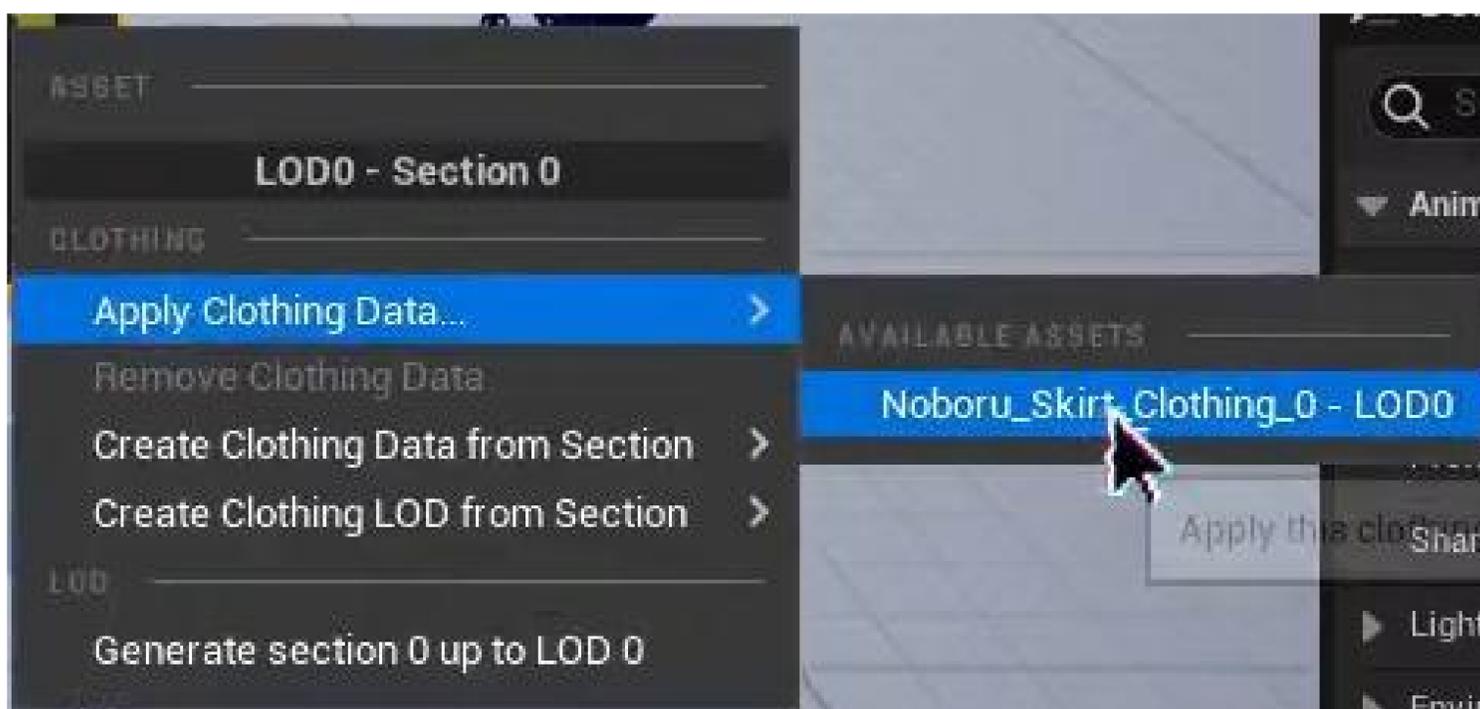


Figura 87. Físicas de Noboru. Cloth Paint de Noboru. Elaboración propia (2024).

Activación y Pintado de Física en la Tela:

- Dar clic en “*Activate Cloth Paint*” y proceder a pintar las áreas que se verían afectadas por las físicas. Este proceso define cómo la tela reaccionará a las fuerzas físicas en el juego.

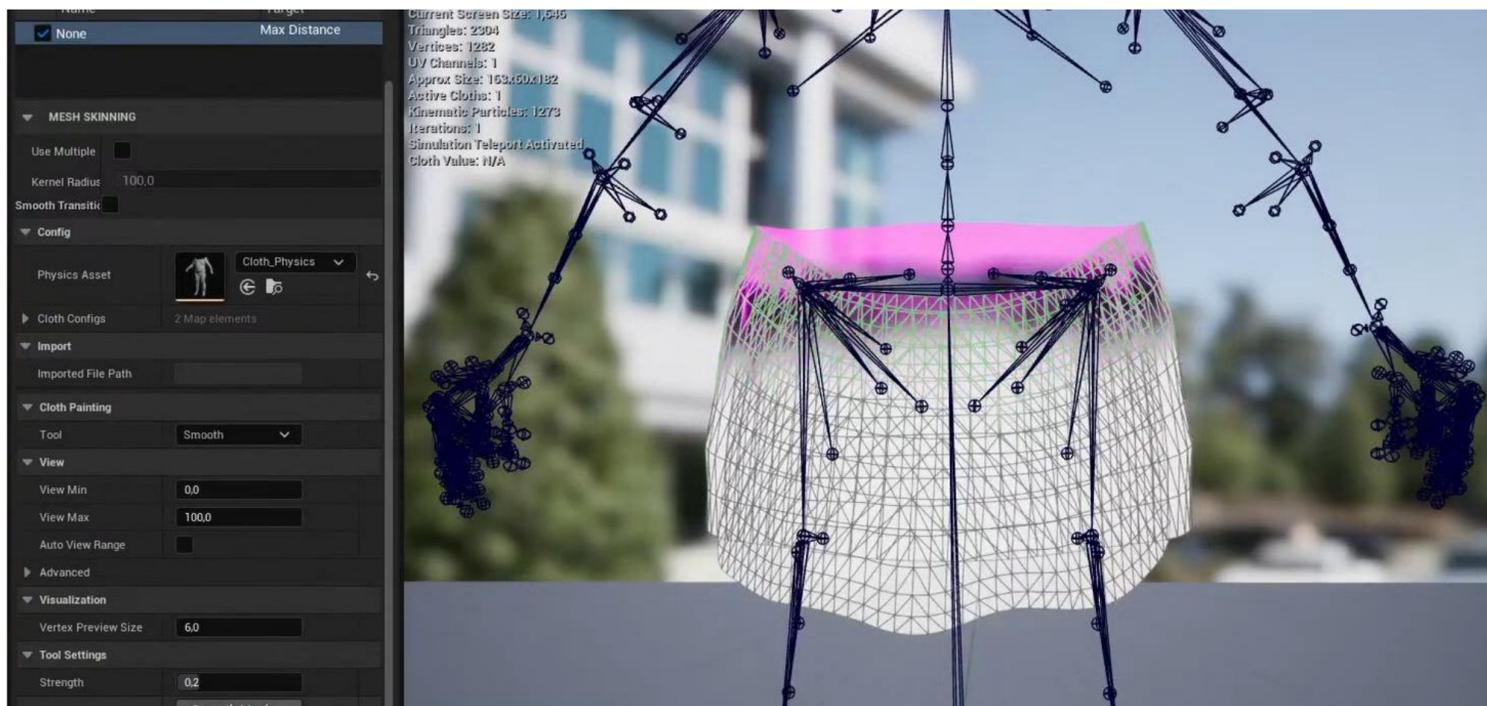


Figura 88. Físicas de Noboru. Cloth Paint de Noboru. Elaboración propia (2024).

Revisión y Ajuste del *Physic Asset*:

- Revisar en *Asset Details* el *Physic Asset* de nuestro *Skeletal Mesh*. El *Physic Asset* es el componente que conforma los colisionadores del cuerpo de nuestro personaje y sobre el cual interactuarán el pelo y la tela del personaje.

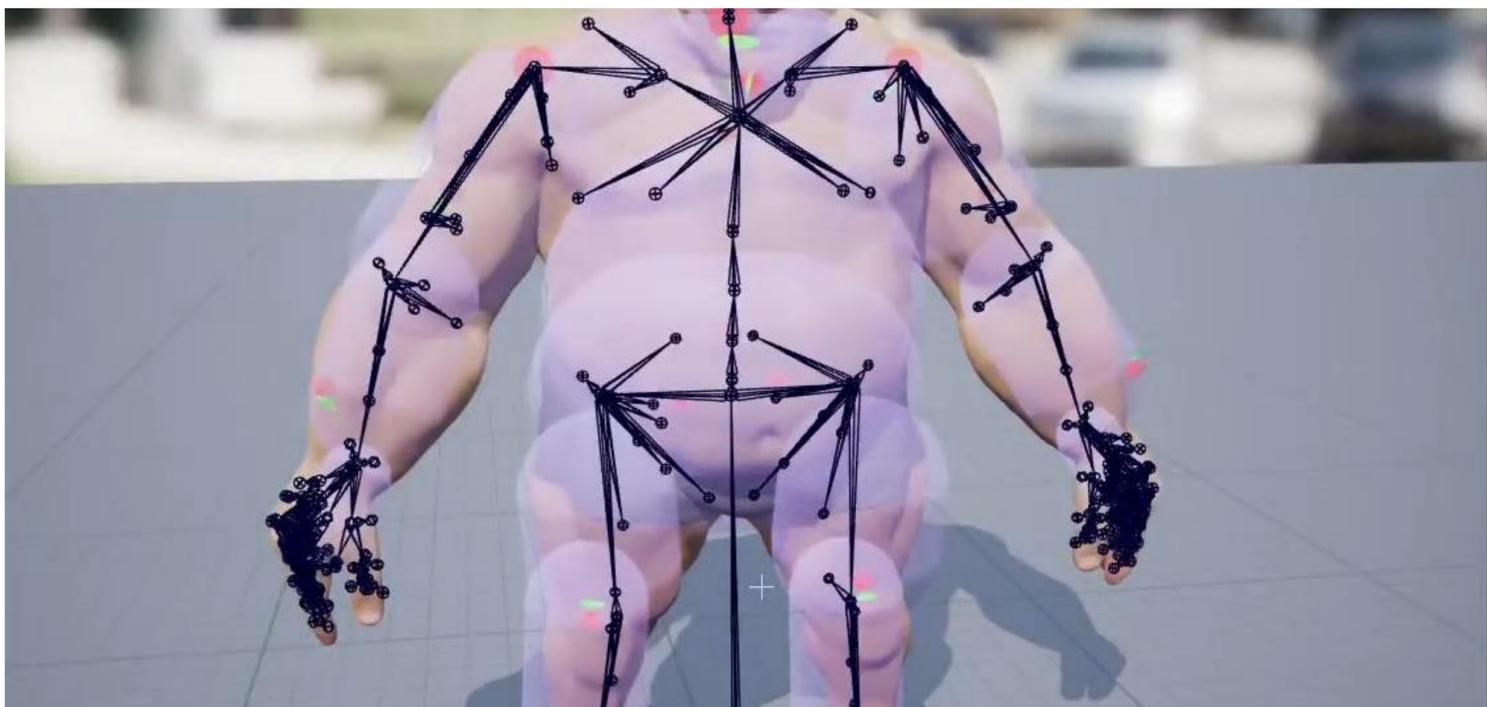


Figura 89. Físicas de Noboru. Physic Asset de Noboru. Elaboración propia (2024).

Estos pasos son esenciales para lograr una simulación realista y eficiente de pelo y la tela en Unreal Engine. Es necesario asegurarse de seguir cada paso con precisión garantizará que las simulaciones funcionen correctamente y se integren bien con los demás componentes del rig del personaje.

2. Configuración de físicas

Este apartado se enfocará en la simulación de piezas de armadura que no pueden ser adecuadamente representadas mediante simulaciones de tela. En el caso específico del personaje diseñado para este estudio, la coraza pélvica, llamada *kusazuri*, típica de la armadura japonesa, necesitó un enfoque distinto debido a su estructura rígida de placas

Simulación de placas conectadas

Para simular correctamente la *kusazuri*, se implementó una serie de *Physics Constraints* conectados a los extremos de cada placa, los cuales permiten que se comporten como una secuencia de elementos conectados. A continuación, se desglosan los pasos detallados:

Creación de *physics constraints*:

- Cada placa de la *kusazuri* debe ser un objeto independiente que se conectará mediante *Physic Constraints*.
- Dentro de *Unreal Engine*, se creó una serie de *Physics Constraints* que conectan los extremos de cada placa. Estas *constraints* definirán cómo interactúan las placas entre sí, permitiendo movimiento pero manteniendo los soportes de la armadura.



Figura 90. Físicas de Noboru. Jerarquía de Static Meshes y Physics Constraints. Elaboración propia (2024).

Padre invisible y emparentado al rig:

- Además de las *Physic Constraints*, se creó un *Static Mesh* de un cubo invisible que funcionará como el padre de la cadena de placas.
- Este cubo invisible se emparenta al hueso más cercano del *rig* (en este caso, el hueso pélvico). Esto asegura que al mover el *rig*, la cadena de *Physic Constraints* también se moverá en consecuencia.
- El cubo invisible permite que las placas de armadura mantengan su estructura mientras caen por su propia gravedad y se deforman de manera realista.

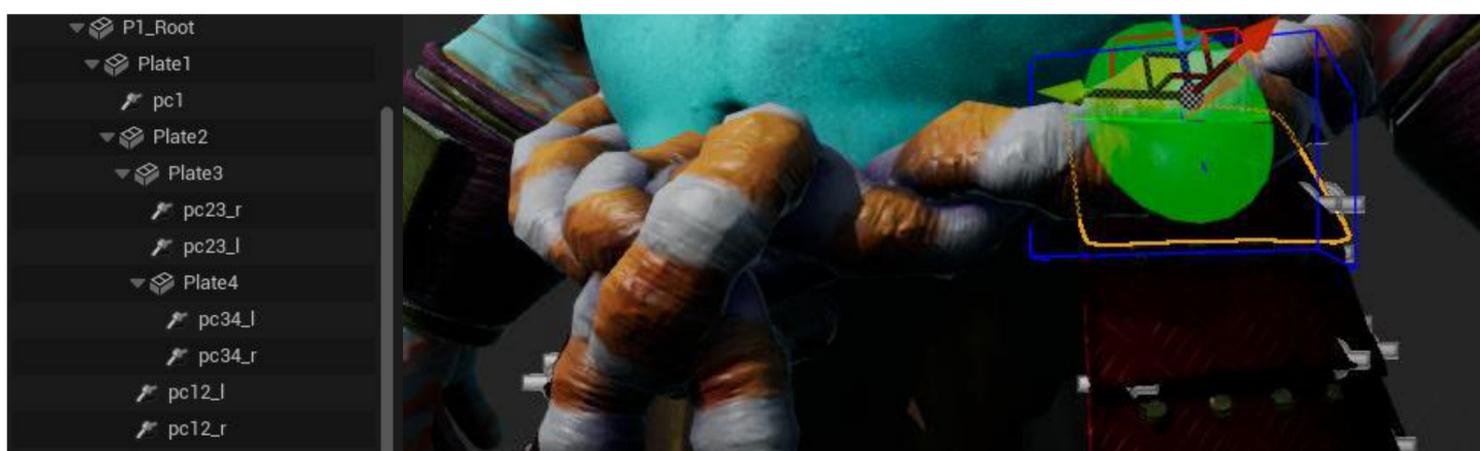


Figura 91. Físicas de Noboru. Jerarquía de Static Meshes y Physics Constraints. Elaboración propia (2024).

Configuración de la física y gravedad:

- En el caso en estudio, se configuraron las propiedades de física y gravedad para cada una de las placas, asegurando que respondan correctamente a las fuerzas en el juego.
- Se ajustaron los parámetros de las *Physics Constraints* para obtener un balance adecuado entre rigidez y flexibilidad, permitiendo un movimiento natural sin que las placas se desalineen o atraviesen entre sí.

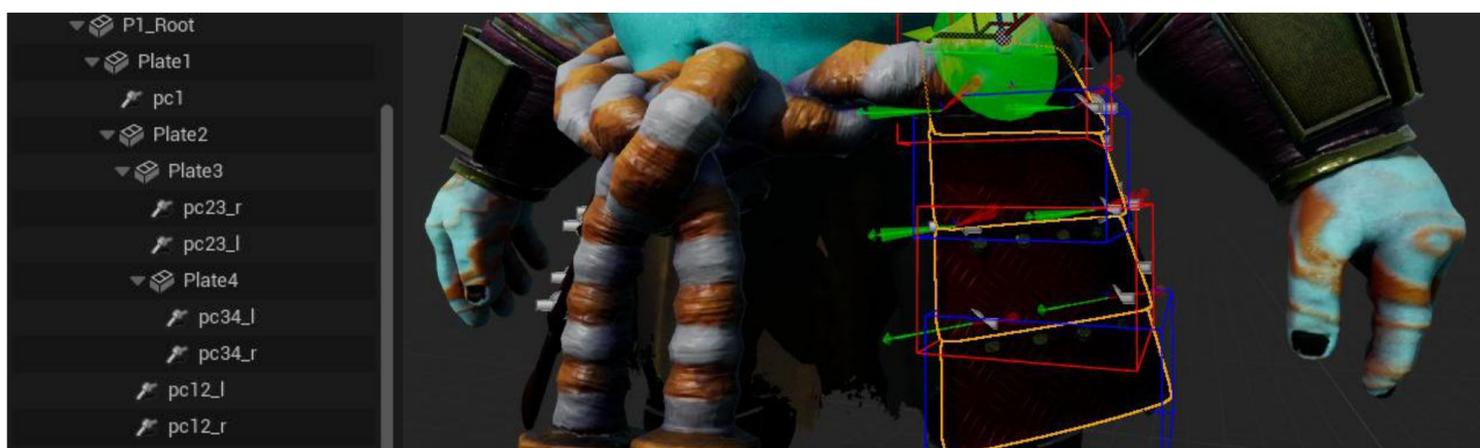


Figura 92. Físicas de Noboru. Jerarquía de Static Meshes y Physics Constraints. Elaboración propia (2024).

Emparentado de la cadena de placas:

- La cadena de *Physic Constraints*, junto con el cubo invisible, se integró en el esqueleto del personaje, específicamente en el área pélvica.
- El cubo invisible y las *constraints* responderán adecuadamente al movimiento del personaje, permitiendo una simulación de armadura realista y cohesiva.

Resultado y ajustes finales

Con estos pasos, la *kusazuri* se comporta como una secuencia de placas conectadas, de modo que cae y se mueve de forma natural según la gravedad y las fuerzas aplicadas por el movimiento del personaje. Este método evita los problemas de deformación inadecuada que surgirían si se intentara simular la armadura como tela.

Finalmente, se realizaron pruebas en el juego para ajustar cualquier problema en la interacción entre las placas y el cuerpo del personaje, asegurándose de que la simulación fuera tanto visualmente impresionante como funcional en términos de *gameplay*.

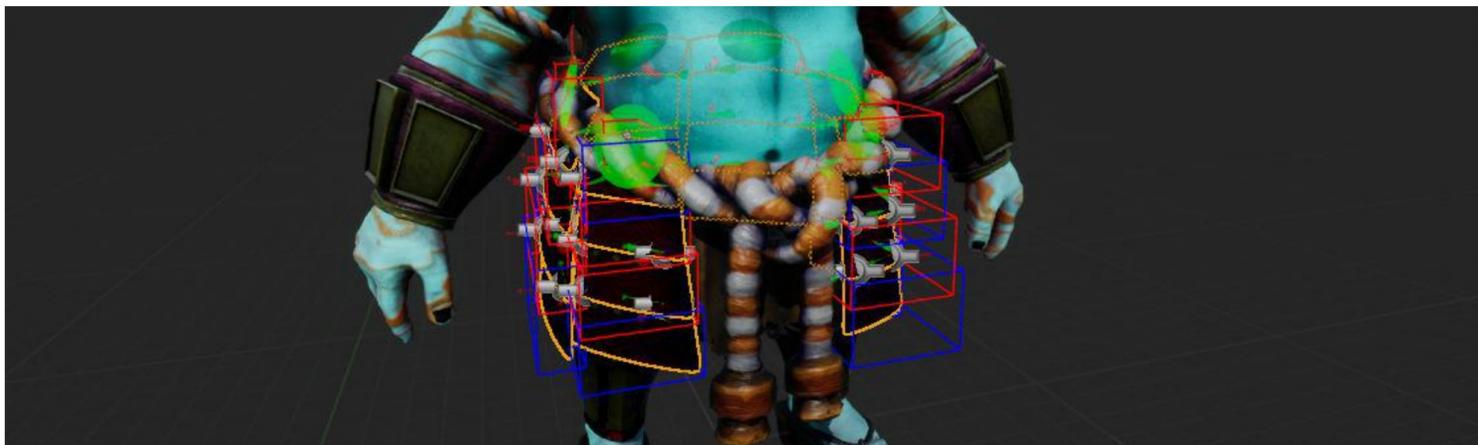


Figura 93. Físicas de Noboru. Jerarquía de Static Meshes y Physics Constraints. Elaboración propia (2024).

COMPARATIVA ENTRE METAHUMAN Y ADVANCED SKELETON

1. Introducción

En este capítulo, se presentará una comparativa detallada entre MetaHuman y Advanced Skeleton, dos herramientas avanzadas de rigging utilizadas para crear personajes animados en la industria de los videojuegos y la animación. Tras haber implementado y optimizado ambos *rigs* en Unreal Engine 5, se explorarán diversos aspectos clave como la preservación estética, tiempos de desarrollo, adaptabilidad, usabilidad, calidad del rig y pruebas de animación, para determinar cuál de estas herramientas ofrece mayores ventajas en un entorno de producción profesional.

2. Preservación estética

Una vez completados ambos *rigs*, es evidente que MetaHuman sobresale en la preservación estética del personaje. La metodología aplicada ha demostrado que esta herramienta no solo crea avatares genéricos e hiperrealistas, sino que también puede personalizarse para mantener la estética y el atractivo del personaje original. Los personajes creados con MetaHuman y Advanced Skeleton son visualmente idénticos, subrayando la capacidad de MetaHuman para adaptarse y potenciar a los personajes estilizados.

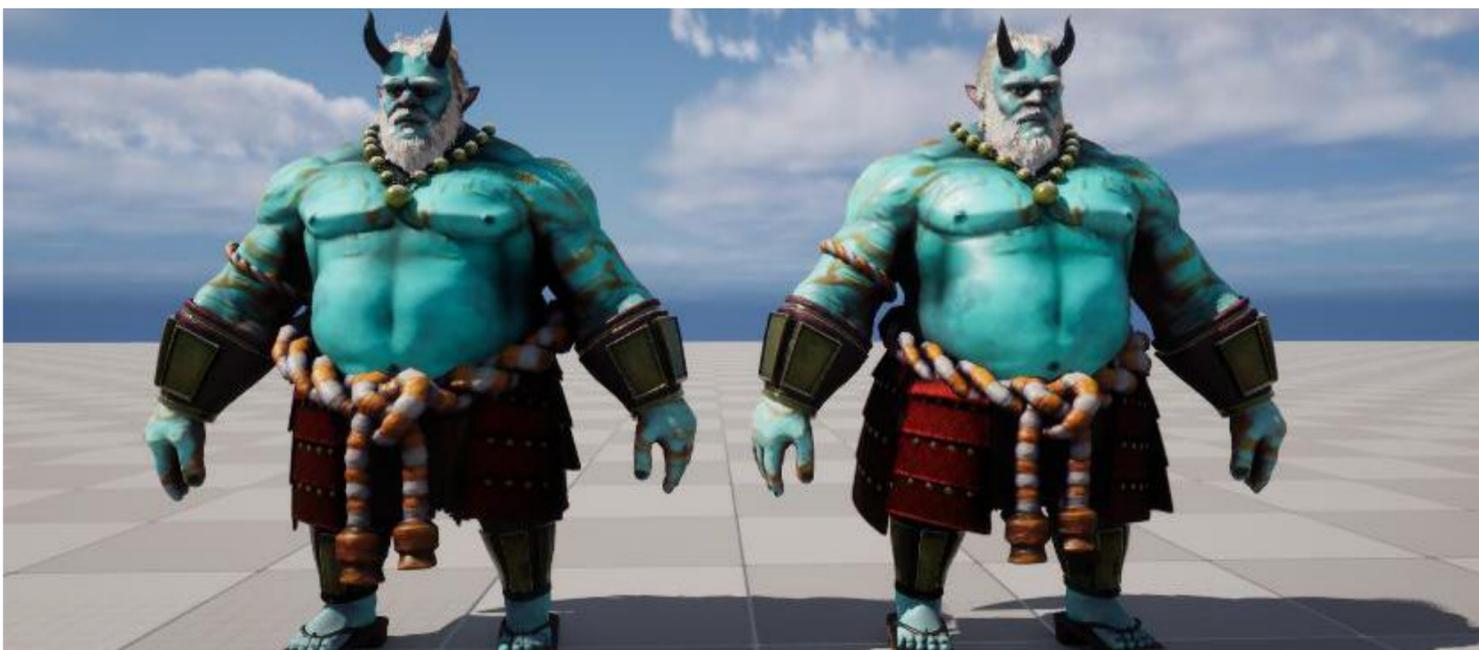


Figura 94. Comparativa de Noboru. Comparativa de rig en MetaHuman y Advanced Skeleton. Elaboración propia (2024).

3. Tiempos de desarrollo

El tiempo de desarrollo para ambos *rigs* fue considerablemente similar debido a los múltiples desafíos y horas de investigación necesarias para superar los inconvenientes técnicos. No obstante, la metodología de Advanced Skeleton puede ser más accesible para un artista 3D típico, ya que el proceso de *rigging* es más directo. En contraste, MetaHuman exige una curva de aprendizaje más técnica que implica una deconstrucción y reconstrucción del rig preexistente.

4. Adaptabilidad de la herramienta

MetaHuman se adapta de manera más eficiente al flujo de trabajo dentro de Unreal Engine debido a su compatibilidad directa con el entorno. Esto incluye acceso a *MetaHuman Animator* y compatibilidad con el rig de *Mannequin*, *Live Link* y *Modular Control Rig*. Por otra parte, Advanced Skeleton está limitado a las capacidades que ofrece el desarrollador, ya que está construido para su adecuado funcionamiento dentro de Autodesk Maya, lo que puede restringir su adaptabilidad en ciertos entornos de producción.

5. Usabilidad de la herramienta

La usabilidad de MetaHuman es más compleja cuando se empieza a utilizar, ya que requiere de conocimientos técnicos avanzados para aprovechar al máximo sus capacidades. Sin embargo, una vez superada la curva de aprendizaje, ofrece una plataforma robusta y versátil para crear *rigs* de alta calidad. Por su parte, Advanced Skeleton, al ser una herramienta creada para su funcionamiento nativo en Autodesk Maya, ofrece una usabilidad más intuitiva para los artistas familiarizados con este software, aunque tiene limitaciones en comparación con las capacidades integradas de MetaHuman en Unreal Engine.

6. Calidad del rig

En términos de calidad del rig, MetaHuman supera notablemente a Advanced Skeleton. El rig de MetaHuman es mucho más complejo y ofrece una mayor gama de expresiones faciales y deformaciones musculares realistas. Además, incluye niveles de detalle (LOD, por sus siglas en inglés) integrados, compatibilidad con captura de movimiento y *rigs* genéricos de Unreal Engine, lo que la convierte en una herramienta superior para la creación de personajes altamente detallados y realistas.



Figura 95. Comparativa de Noboru. Comparativa de rig en MetaHuman y Advanced Skeleton. Elaboración propia (2024).

7. Pruebas de animación

Las pruebas de animación revelaron que MetaHuman, gracias a su compatibilidad con rigs genéricos de Unreal Engine y a la captura de movimiento facial, simplifica considerablemente el proceso de animación, permitiendo resultados de calidad AAA con menos esfuerzo. Aunque Advanced Skeleton facilita el trabajo de los animadores experimentados en Autodesk Maya, los procesos pueden ser más prolongados y laboriosos en comparación con la eficiencia de MetaHuman dentro de Unreal Engine y sus tecnologías innovadoras.

8. Conclusiones

En conclusión, aunque MetaHuman es una herramienta relativamente nueva y su uso y personalización requieren un extenso estudio, los resultados obtenidos son comparables, y a menudo superiores, a los logrados con herramientas de *auto-rigging* como Advanced Skeleton. Por lo tanto, es altamente recomendable adaptar el pipeline de trabajo para incorporar MetaHuman, aprovechando sus capacidades avanzadas y su integración directa con Unreal Engine para producir personajes de alta calidad de manera más eficiente.

MetaHuman ha demostrado ser más que una herramienta para crear avatares genéricos. Su capacidad para preservar la estética y el carácter único de los personajes estilizados la convierte en una solución poderosa y versátil para la industria de la animación y los videojuegos. Esta comparativa evidencia que MetaHuman puede satisfacer tanto las demandas estilizadas como las técnicas, proporcionando una plataforma robusta que puede simplificar y mejorar significativamente el proceso de creación de personajes.

Además, MetaHuman facilita la adaptación a las necesidades de producción contemporáneas reduciendo los tiempos de desarrollo sin comprometer la calidad. Esto, combinado con sus avanzadas capacidades de *rigging* y compatibilidad con herramientas de captura de movimiento y animación, hace de MetaHuman una adición valiosa y recomendable para cualquier *pipeline* de trabajo en la industria del entretenimiento digital.





METAHUMAN UNLOCKED



Universidad
LCI VERITAS

Escuela de
Animación Digital

Autor: Letner Hernández Gómez

Tutora: Verónica Morera Carballo

Universidad LCI Veritas

Costa Rica, 2024

