

UNIVERSIDAD POLITÉCNICA DE VALENCIA

ESCUELA POLITÉCNICA SUPERIOR DE GANDÍA **Máster en Postproducción Digital**



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA POLITÈCNICA
SUPERIOR DE GANDÍA

Diseño e integración de modelo 3D sobre imagen real de vídeo.

TRABAJO FINAL DE MASTER

Autor/a:

Ortega Zamora, Kevin Wilson

Tutor/a:

García Miragall, Carlos Manuel

GANDÍA, 2021

Resumen.

El presente trabajo procura abordar el proceso completo para la creación e integración realista de un elemento generado por ordenador sobre un vídeo. Abordaremos la creación del objeto 3D, que comprende su modelado en un software de modelado 3D, su despliegue en mapas UV y posterior texturización mediante técnicas PBR, luego pasaremos a la técnica de *matchmoving* para el rastreo de cámara y por el proceso de iluminación del objeto mediante el uso de imágenes HDRi tomadas del entorno real y su renderizado mediante motor de render, para finalmente, pasar a la etapa de composición e integración con el vídeo haciendo uso de técnicas de postproducción con After Effects.

Palabras clave.

Modelado, texturizado, imagen HDR, PBR, *matchmoving*, composición.

Abstract.

The present work tries to approach the complete process for the creation and realistic integration of a computer-generated element on a video. We will address the creation of the 3D object, which includes its modeling in a 3D modeling software, its display in UV maps and subsequent texturing using PBR techniques, then we will move on to the matchmoving technique for camera tracking and the object's illumination process. by using HDRi images taken from the real environment and rendering them using a rendering engine, to finally move on to the stage of composition and integration with the video using post-production techniques with After Effects.

Keywords.

Modeling, texturing, HDR image, PBR, matchmoving, compositing.

Agradecimientos

Mis mas sinceros agradecimientos a mi familia y novia por el apoyo emocional brindado durante la realización de este proyecto.

Agradecimientos también a Carlos Pérez por los consejos profesionales y material técnico prestado.

Y sobre todo agradezco a la universidad por los conocimientos que me han brindado y a mi tutor, Carlos Manuel Garcia Miragall, por la guía y ayuda constante que me brindo durante todo el proceso de realización de este trabajo.

Índice

1. Introducción	7
1.1 Motivación	7
1.2 Objetivos	7
1.3 Metodología	7
1.4 Estructura de la memoria	8
2. Contexto	9
2.1 Modelado digital 3D	9
2.1.1 Flujo de trabajo en la producción 3D	9
2.1.2 Preparación para modelar	12
2.1.3 Fundamentos y conceptos del modelado digital	12
2.1.4 Modelos digitales	16
2.1.5 Softwares de modelado 3D	18
2.2 Texturizado PBR (Physically Based Rendering)	18
2.2.1 Flujo de trabajo PBR	19
2.2.2 Tipos de texturas	19
2.2.3 Softwares de texturización PBR en tiempo real	22
2.3 Matchmoving	22
2.3.1 Flujo de trabajo de Matchmoving	23
2.4 Imágenes HDR (HDRi)	25
2.4.1 Rango dinámico	25
2.4.2 ¿Qué es una imagen HDR?	25
3. Realización	26
3.1 Preproducción	26
3.1.1 Referencias	26

3.2 Producción	27
3.2.1 Modelado de Iron Giant	27
3.2.2 Mapeado UV	33
3.2.3 Texturizado en Substance painter	37
3.2.3.1 Importar malla a Substance painter	37
3.2.3.2 Bake de high poly sobre low poly	39
3.2.3.3 Aplicando smart materials y texturas.	42
3.2.3.4 Exportando texturas.	47
3.2.3.5 Importando texturas Blender.	48
3.2.4 Matchmoving de material rodado	50
3.2.4.1 Evaluación de material rodado	50
3.2.4.2 Aplicación de información	51
3.2.4.3 Definición de cámara	51
3.2.4.4 Montaje del set	53
3.2.4.5 Evaluación del matchmoving	54
3.2.4.6 Entrega de escena	54
3.2.5 Iluminación de escena	54
3.2.5.1 Creación de imagen HDR	55
3.2.5.2 Importación de imagen HDR a Blender	57
3.2.5.3 Iluminación con sun	57
3.2.6 Render de escena	58
3.2.6.1 Configuración del Render	58
3.2.6.2 Configuración del formato de salida	59
3.2.6.3 Envío de escena a render	62
3.3 Postproducción	62
3.3.1 Importación de archivos a After Effects	62

3.3.2 Creación y configuración de composición	63
3.3.3 Preparación de archivo OpenEXR MultiLayer	64
3.3.4 Colorización de Iron Giant	65
3.3.5 Exportación de composición	66
4. Conclusiones	67
5. Referencias	68

1. Introducción

1.1. Motivación

Mi interés en la realización de este proyecto nace de la necesidad de poner en práctica algunos de los conocimientos que he podido recabar a lo largo de mi formación como estudiante y profesional, como son: el modelado 3D, la texturización, renders basados en física (PBR¹), *matchmoving*², colorización, etc., en un proyecto donde pueda integrar todas estas técnicas y conseguir un resultado profesional. También me parecía una buena oportunidad para dirigirme hacia el mundo de los efectos visuales y producción 3D, explorar algunas alternativas gratuitas a los *softwares* estándar 3D más posicionados, como lo es Blender³, y poder generar un flujo de trabajo eficiente que se adecue a mis recursos y necesidades.

1.2 Objetivos

El objetivo principal de este proyecto es utilizar varias técnicas empleadas en la industria del cine y tv de forma ordenada y sistemática para lograr integrar un modelo 3D sobre un vídeo de forma convincente.

Objetivos secundarios:

- Completar la creación de un modelo 3D en base a referencias en todas sus fases, el modelado, despliegue de UVs⁴ y texturización
- Generar un *matchmoving* convincente
- Integrar modelo 3D sobre video real

1.3 Metodología

La fase inicial de este proyecto consiste en la búsqueda de material bibliográfico y recursos de video tutoriales sobre los temas de modelado digital, despliegue de mapas UV, texturización PBR, *matchmoving*, imágenes HDR⁵ y composición.

La segunda fase comprende el estudio y comprensión del material recopilado y la búsqueda de referencias. Con esta información empezamos la preproducción donde se estructura el flujo de trabajo y *softwares* con los que se desarrolla el audiovisual.

¹ *Physically based rendering* o representación basada en física es un flujo de trabajo de texturización de materiales 3D que reacciona a cualquier condición de luz y genera resultados hiperrealistas.

² Técnica para la creación de una cámara digital a partir de la información de una cámara real emulando todos sus movimientos y características.

³ Es una suite de creación 3D gratuita y de código abierto cofundada por Tom Roosendaal en 1995.

⁴ Las letra U y V hacen referencia a los dos ejes del 2D ya que X, Y y Z están reservadas para el 3D

⁵ *High Dynamic Range* o alto rango dinámico es una técnica en la que se fusionan varias fotos de diferente exposición para conseguir una foto con la mayor cantidad de información y detalle.

Como tercera fase se procede a la producción o realización del material de vídeo donde se colocará el modelo y la toma de fotografías HRDi para la iluminación realista del entorno 3D. Se empieza el modelado 3D, sus texturas y posado. Luego se empieza la realización del *matchmoving* para insertar el modelo generado y se ilumina la escena con la imagen HDRI para enviar a *render*.

Con este material inicia la fase de postproducción donde empieza el tratamiento del material de video y *renders* generados para mediante composición lograr la integración del modelo 3D sobre la imagen de video real.

La fase final comprende la valoración de los conocimientos adquiridos más su redacción en un informe escrito de todos los pasos realizados para obtener el resultado final deseado y las conclusiones a las que hemos podido llegar durante todo el proceso.

1.4 Estructura de la memoria

En la primera parte de esta memoria se detallan la introducción, motivación, objetivos y metodología empleada para su finalización.

La segunda parte de la memoria hace una recopilación de fundamentos teóricos necesarios para el cumplimiento de los objetivos mencionados.

Luego en la tercera parte de la memoria encontraremos el desarrollo práctico de estos fundamentos teóricos aplicados al proyecto propuesto.

Finalmente, en el último apartado comentaremos las conclusiones, valoraremos el cumplimiento de los objetivos, a nivel personal que ha significado la realización de este proyecto y mencionaremos las referencias bibliográficas en las que se ha referenciado este trabajo.

2. Contexto

En este apartado encontraremos las diferentes técnicas y conceptos que han sido analizados e implementados para la adecuada ejecución de este proyecto, tomando en cuenta sus antecedentes históricos y ejemplos prácticos en el cine o televisión.

2.1 Modelado digital 3D

“El modelado digital se refiere al proceso de crear una representación matemática de una forma tridimensional de un objeto. El resultado de esta creación es lo que la industria llama modelo 3D o malla 3D”. (Vaughan, 2011)

El modelado 3D generalmente empieza con la conceptualización en dos dimensiones del modelo y su posterior desarrollo mediante un *software* de modelado, aunque existen otras alternativas como la fotogrametría o escaneo láser de un objeto real para su conversión digital.

2.1.1 Flujo de trabajo en la producción 3D

Un flujo de trabajo comprende un número determinado de tareas que deben ser ejecutadas en un orden específico previamente determinado adaptado a los requerimientos del proyecto, similar a la línea de ensamblaje de automóviles, para conseguir un producto final de calidad optimizando tiempo y recursos. El flujo de trabajo variará de acuerdo con el objetivo final deseado, por ejemplo, en un corto animado se requerirá un flujo de trabajo que, una vez terminado el modelado, tenga tareas de texturizado, *rigging*¹, animación, iluminación y render. Por otra parte, si el producto final deseado es una figura de acción impresa en 3D, todas estas tareas previamente mencionadas serán omitidas.

El flujo de trabajo en una producción 3D se divide en tres etapas: Preproducción, Producción y Postproducción. Para fines prácticos usaremos el flujo de trabajo estándar de una producción de un corto animado (ver fig. 1).

¹ Proceso en una producción 3D que consiste en la creación de un esqueleto para un personaje 3D u objeto que deseemos animar.

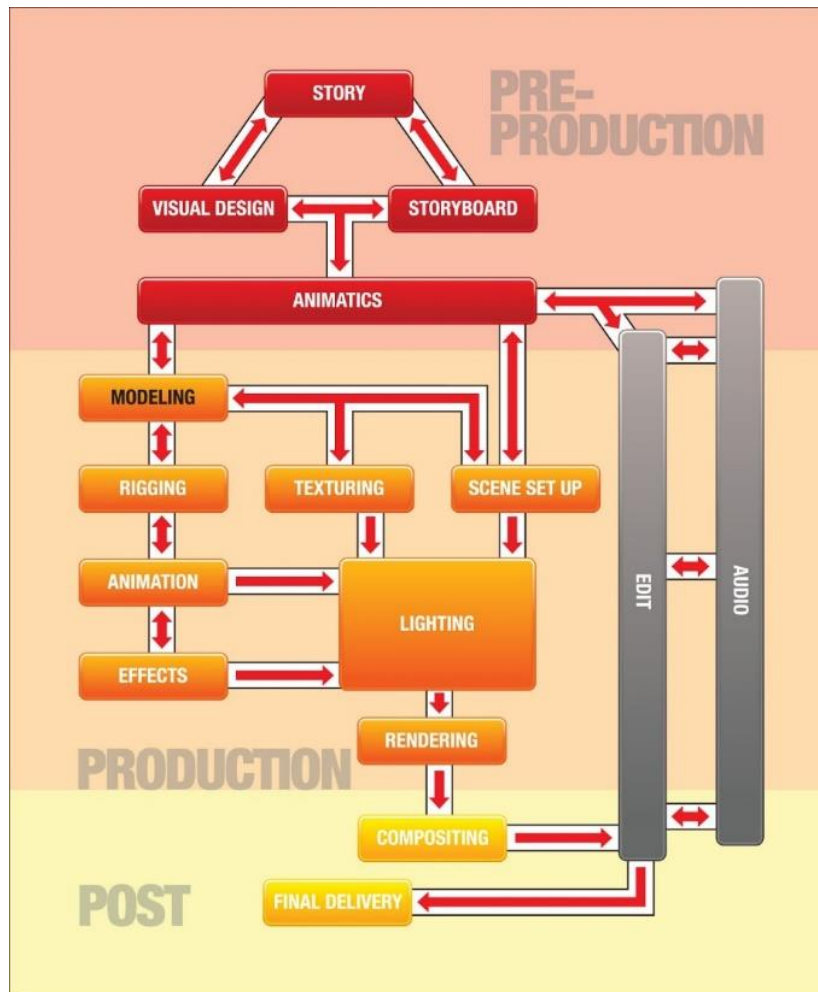


Figura 1. Flujo de trabajo estándar en animación 3D.

- Preproducción

La preproducción es la etapa de preparación de todos los elementos involucrados en la producción y es la base del proyecto. En esta etapa se desarrolla la historia, el aspecto visual del corto y la planificación para la etapa de producción. Es una etapa muy crítica que influirá mucho en si un proyecto es exitoso o no.

- Guion: Es donde se crea y conceptualiza la historia, ambientación, personajes, diálogos, etc.
- Diseño visual: En esta etapa se crea el *concept art* que definirá la estética visual de todos los elementos involucrados en la historia.
- *Storyboard*: En esta parte se definen, mediante ilustraciones, los planos y tiros de cámara que permiten tener una previsualización de la historia.

- *Animatic*: Es una representación animada en 2D, con animación sencilla, de la totalidad del filme.

- **Producción**

La etapa de producción es donde se crearán los elementos, previamente conceptualizados en la fase de preproducción, que formarán parte de la animación final. En esta etapa el modelador digital juega un rol de alta relevancia dentro del flujo de trabajo y debe mantenerse fiel a los parámetros previamente establecidos por el director y el departamento de preproducción.

- Modelado: Donde se crea la malla 3D.
- *Rigging*: creación de esqueleto para animación.
- Texturizado: creación de materiales y texturas sobre las mallas 3D.
- Configuración de escenas: disposición de elementos 3D en el entorno virtual, decorado del entorno en base al concepto artístico.
- Animación: animación de personajes, objetos, cámaras, etc.
- Efectos: creación de fluidos, fuego, humo, partículas
- Iluminación: iluminación del entorno mediante luces virtuales o imágenes HDR para dar el toque final antes de renderizar.
- *Render*: exportación en diferentes pases de *render* para su refinamiento en la postproducción.

- **Postproducción**

La Postproducción es la fase final del flujo de trabajo que comprende la recopilación del material generado en la etapa producción para su refinamiento. Puede ser una de las etapas que más tiempo consume debido a que este determina el resultado final del corto.

- Composición: se combinan los *renders* individuales para crear una escena mediante técnicas de corrección de color, efectos de profundidad, etc.
- Audio: desarrollo de música, efectos sonoros y masterización.

- Montaje: edición del material generado usando de guía el *animatic* con cambios en edición si se cree necesario.

2.1.2 Preparación para modelar

William Vaughan considera que existen 3 herramientas básicas al momento de empezar a modelar y generar un resultado fotorrealista. (Vaughan, 2011, p. 78).

- **Referencias:** Las referencias consisten en la recopilación de imágenes guía de alta calidad u objetos reales que sirvan de guía para lo que se va a modelar, cuanto mayor sea el número y calidad de las referencias, mayor será el grado de fotorrealismo alcanzado por el artista.
- **Observación:** La observación consiste no sólo en que el objeto modelado coincida en forma al real, sino comprender qué características hacen que el objeto luzca como real y emularlas.
- **Resolución de problemas:** Esta herramienta consiste en la visualización del problema antes de empezar a ejecutar el modelado, de esta manera podremos determinar una estrategia o técnica a emplear optimizando en tiempo y trabajo.

2.1.3 Fundamentos y conceptos del modelado digital

En este apartado nos centraremos en identificar los componentes presentes dentro de un modelo 3D para tener un mejor entendimiento de que comprende el modelado digital. (Vaughan, 2011, p. 100 - 112).

1. Puntos o vértices: Son los elementos más básicos que componen el desarrollo de un modelo 3D. Estos puntos existen en un espacio 3D definido por las coordenadas X, Y y Z. Los puntos carecen de alto, ancho y profundidad por lo que no pueden ser renderizados.

La unión de 2 puntos genera una arista y la unión de 3 o más puntos genera un polígono (ver fig. 2).

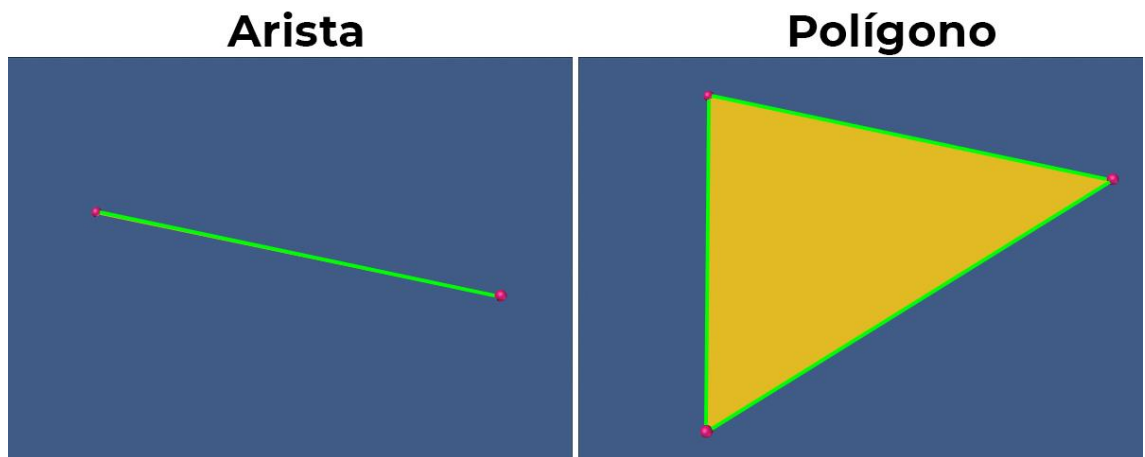


Figura 2. Ejemplo de arista y polígono

2. Mapa de Vértices – Vertex Maps: Los puntos individualmente tienen la capacidad de guardar información de su posición y rotación dentro del entorno 3D, por otro lado, en conjunto estos puntos tienen la capacidad de guardar información adicional por medio de mapa de vértices. Los mapas de vértices son la totalidad de puntos de la malla trabajando en conjunto para guardar información adicional.

Tipos de mapas de vértices:

- o **Textura (UV):** Los mapas UV o de texturas son los mapas de vértices más comunes, estos almacenan información sobre la ubicación de la textura agregando dos coordenadas adicionales a los puntos del objeto, eje U (horizontal) y V (vertical), que se ejecutan a través de un plano bidimensional sobre el cual se puede luego pintar una textura plana, por lo tanto podemos decir que las coordenadas UV son una representación 2D del espacio 3D que establecen una relación entre la imagen bidimensional y la superficie tridimensional a la que se aplicará la imagen (ver fig. 3).

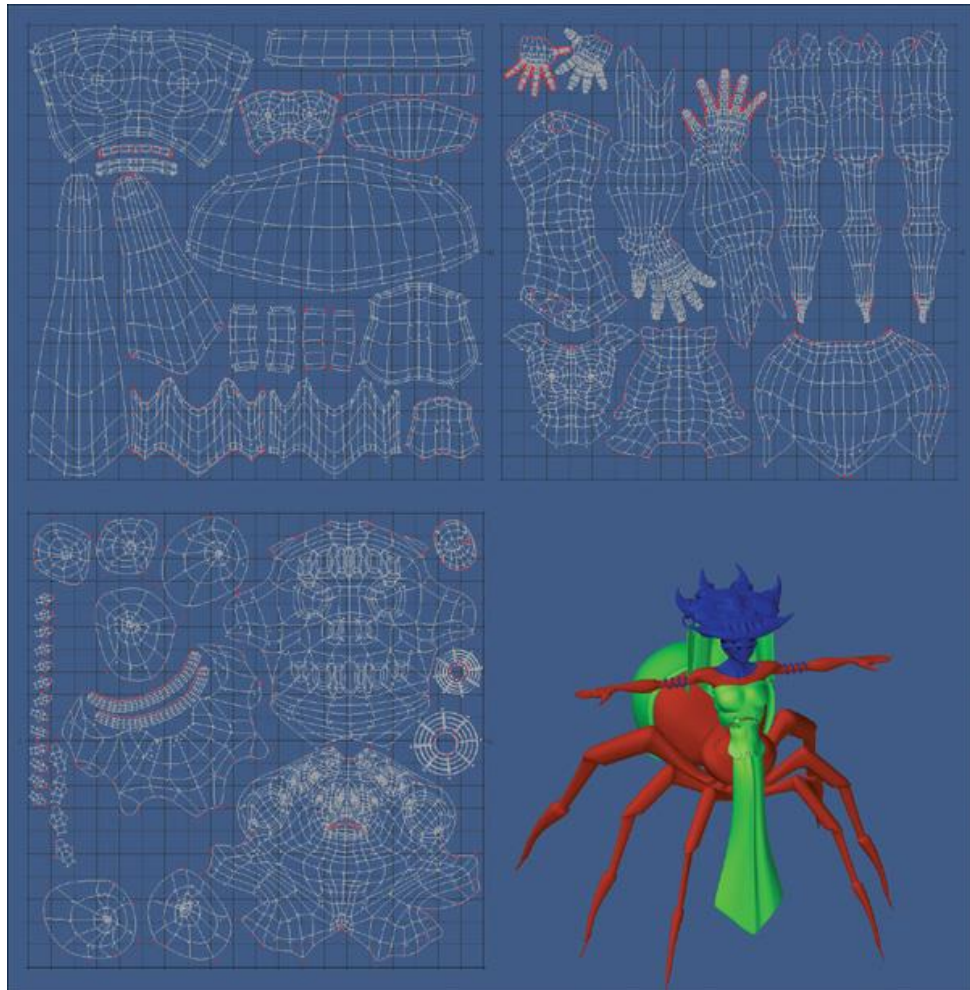


Figura 3. Tres mapas UV asignados a un mismo modelo (UDIMs) (Vaughan, 2011, p. 104).

Los mapas UV presentan una desventaja al momento de texturizar una malla y es que al ser solo un mapa de textura para la totalidad de la superficie 3D esta puede quedar corta para proyectos en los que se requiera de mayor detalle y calidad más aún si el modelo consta de muchas partes y detalles. Por lo general esto ocurre en proyectos para cine y animación que requerirán de una mayor calidad y resolución de texturas, para los demás campos como, por ejemplo, videojuegos, visualización arquitectónica, etc., suele ser suficiente el uso de un solo mapa de textura. UDIM, que significa *U DIMension*, ofrece la posibilidad de poder trabajar con varias texturas UV para una misma malla en donde se puede repartir partes de la malla para aumentar el nivel de detalle y calidad de las texturas (ver fig. 3).

- **Weight – peso:** Estos mapas almacenan información de un valor fijado generalmente entre -100 y 100. Su aplicativo más común es para definir el grado de influencia de un *bone* (hueso) sobre los puntos al momento de generar un *rigging* y deformar la malla en animación. Otros posibles aplicativos son para generar máscaras sobre superficies al momento de texturizar, determinar el grado de influencia de una simulación dinámica (*Dynamic simulation*), ayudar el proceso de modelado, etc.

- **Morph:** Los mapas de *morph* almacenan información de desplazamiento de la posición de un punto.

- **Color:** Los mapas de color almacenan información de 4 canales, rojo, verde, azul y alpha (RGBA). Es usado para dar color a la malla o también para crear canales de selección al momento de postproducir un render.

- **Selección:** El mapa de selección, también conocido como *selection set* (conjunto de selección), almacena información de un solo estado del punto, este estado puede ser seleccionado o no seleccionado, es decir, que contiene información que define cuales puntos de la malla se encuentran seleccionados y cuáles no, que permite al modelador recuperar rápidamente un grupo definido de puntos de la malla para su edición o afectación por simulaciones dinámicas.

3. Aristas – Edges: Una arista es una línea unidimensional que conecta dos puntos en un polígono

4. Edge weights: El *edge weight* incrementa o disminuye la nitidez de una arista entre dos polígonos de subdivisión de superficie, permitiendo mayor dureza o suavidad en las esquinas sin la necesidad de incrementar geometría. El problema principal con los *Edge weights* es que no existe un formato de

transferencia común entre todas las aplicaciones 3D, por lo que los modeladores prefieren usar más geometría en lugar de esta técnica.

5. Curvas – Splines: Es una curva definida por al menos 2 puntos dentro del espacio 3D. El *spline* más utilizado en el modelado digital es la curva de Bézier que se utilizan para modelar curvas suaves utilizando muchos menos puntos de los requeridos en el modelado poligonal.

2.1.4 Modelos digitales

Ahora que conocemos los fundamentos del modelado digital nos centraremos en profundizar en los tipos de modelado digital que existen.

Los modelos digitales se dividen en 3 tipos:

1. Modelo Poligonal: se compone de un conjunto de puntos, aristas y polígonos. Los polígonos, también conocidos como caras (*faces*) son formas geométricas compuestas por un número determinado de puntos que definen la superficie del objeto 3D. Los polígonos compuestos por 3 puntos se llaman triángulos o tris, si está compuestos por 4 puntos se conocen como *quads* y si tiene más de 4 puntos adquiere la nomenclatura *n-gon* siendo n el número de puntos que lo conforman (ver fig. 4).

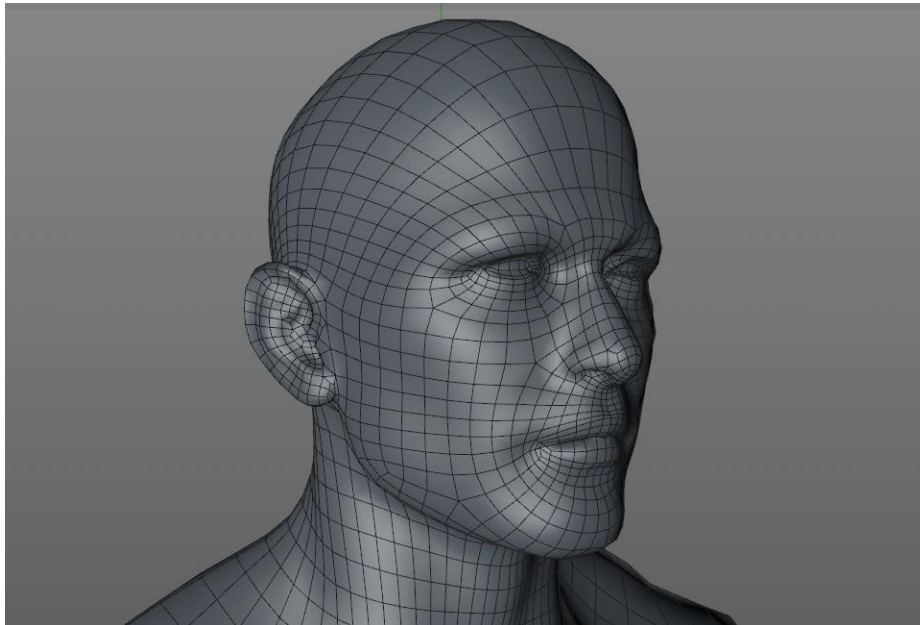


Figura 4. Modelo poligonal de humano.

2. Superficies Nurbs (Non-uniform Rational B-splines): consisten en una malla suave definida por una serie de curvas (*splines*) conectadas. Esta superficie suave se convierte en polígonos al momento de ejecutar un render, por lo que pueden contener una cantidad variable de polígonos (ver fig. 5).

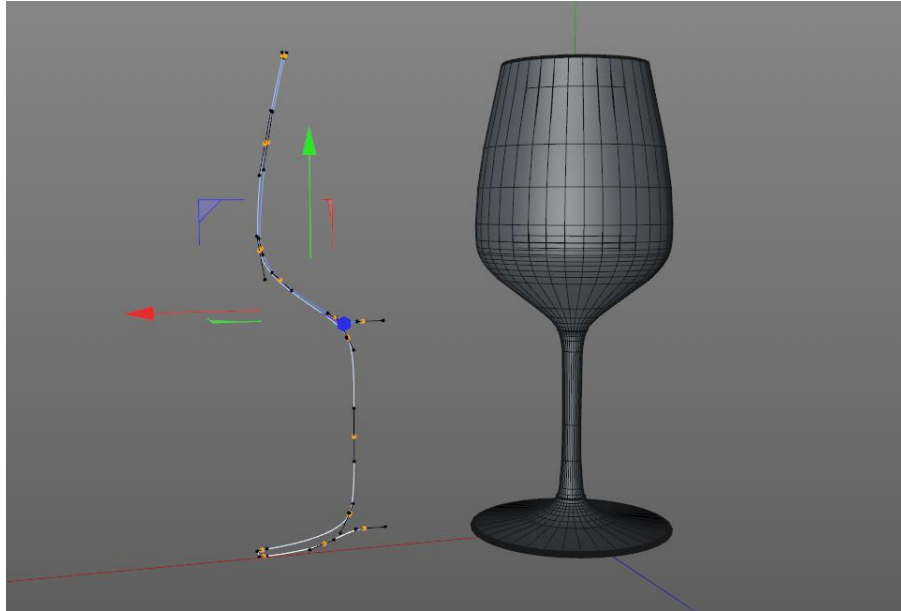


Figura 5. Superficie Nurbs definida por una curva (spline).

3. Superficies de subdivisión: es un tipo de modelado digital que usa un algoritmo de refinamiento para crear superficies curvas suaves a partir de una malla base. Este proceso toma los vértices de la malla base como puntos de control, conocido como control *cage* (jaula de control) para crear una superficie lisa (ver fig. 6).

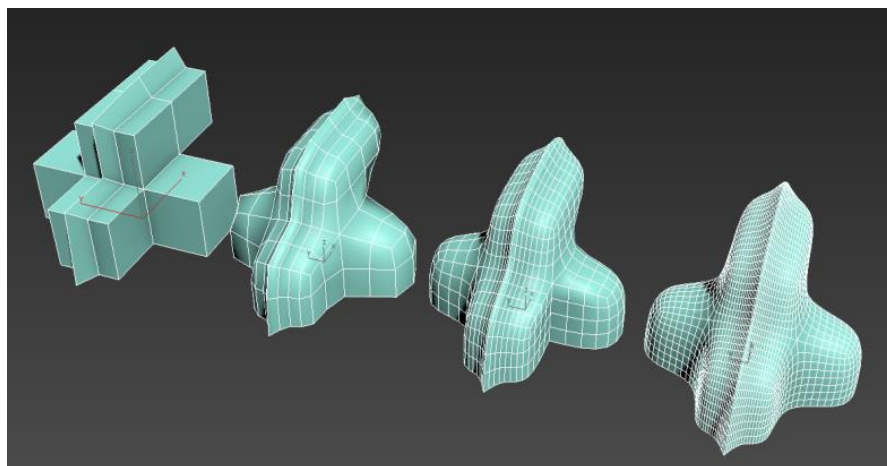


Figura 6. Superficie de subdivisión

2.1.5 Softwares de modelado 3D

Algunos de los softwares más extendidos, en la industria del cine y videojuegos, para el modelado en 3D son: Maya, 3D Studio Max, Cinema 4D, Zbrush, Blender, Modo, etc.

En el presente trabajo usaremos Blender 2.91 para la creación del modelo 3D y despliegue de mapas UV, aunque este consta de otros entornos como el de animación, *shading*, *rendering*, composición, *motion tracking*, *sculpting*, *texture Paint*, *scripting*, etc

Blender

Blender¹ es una suite de creación 3D gratuita y de código abierto cofundada por el neerlandés Tom Roosendaal en 1995 y distribuida de manera gratuita desde 1998. Blender soporta la totalidad del flujo de trabajo de una producción 3D: modelado, montaje, animación, simulación, renderizado, composición, *tracking*² de movimiento además de flujo de trabajo 2D

2.2 Texturizado PBR (Physically Based Rendering)

Abhishek Kumar explica que “un flujo de trabajo de renderizado basado en la física (PBR) juega un papel muy importante en los gráficos modernos, específicamente en los video juegos. Generalmente, la ciencia de la textura y la creación de material 3D es una mezcla de mapas de textura y propiedades de la superficie” (Kumar, 2020) (ver fig. 7).

Con un enfoque PBR, podemos crear resultados perfectos, más precisos y súper realistas que los motores de render de juegos aceptan fácilmente. La representación de gráficos en juegos requiere un algoritmo complejo para lograr el mejor resultado.

Usando PBR podemos minimizar la complejidad del algoritmo. PBR utiliza esencialmente el comportamiento físico de la luz para reproducir materiales de apariencia natural, porque las simulaciones del mundo real cambian con las condiciones de la luz y la física.

¹ <https://www.blender.org/>

² Seguimiento



Figura 7. Material con sus respectivos mapas para texturas en videojuegos.

2.2.1 Flujo de trabajo PBR

En el flujo de trabajo PBR se debe de tener modelada una malla *high poly* rica en detalles más su contraparte *low poly* de baja densidad de polígonos con sus mapas uv desplegados, luego en un software de horneado (*bake*) se procede a convertir los detalles del *high poly* en mapas de normales que se proyectaran sobre el *low poly*, además del mapa de normales en el proceso de horneado se puede generar otro tipos de mapas que ayudarán al momento de pintar las texturas, después en el *software* de texturizado se procede a pintar sobre la malla y una vez concluido la texturización se exportan las texturas configurando los materiales para el motor de render que se utilizará en la etapa de renderizado (Udemy, 2021).

2.2.2 Tipos de texturas

“La textura se refiere a la sensación tangible inmediata de una superficie y la impresión visual de estímulos simples como el color, la orientación y la intensidad en una imagen” (Kumar, 2020). Existen muchos tipos de texturas según los requisitos del 3D y el uso de estos dependerán de los requerimientos del proyecto además del tipo de texturas que maneje el motor de render o videojuegos y el flujo de trabajo del estudio.

Algunos de los tipos de mapas más comunes son:

Diffuse: los diffuse maps representan el color general del modelo; este es el tipo de textura más común. Una imagen de mapa de bits, capturada con una cámara, pintada o escaneada, se puede utilizar como un mapa difuso para representar la calidad fotorrealista de un objeto 3D. Diffuse tiene la información de luz y sombra (ver fig. 8).

Albedo: un albedo map es similar a un diffuse map con la diferencia de que se le ha eliminado toda la información de iluminación y sombra y es literalmente los valores de color de la imagen (ver fig. 8 - 9).



Figura 8. Mapa de diffuse y albedo

Opacity o transparency: un mapa de transparencia depende de los valores de gris de una imagen. Por lo general, las partes negras serán opacas y las partes blancas serán visibles.

Specular: un mapa especular define el brillo de la superficie en una determinada posición.

Bump: se utiliza un mapa de relieve para crear el relieve en la superficie. Podemos usar un mapa de relieve para generar la ilusión de profundidad en la superficie del modelo 3D. Un mapa de relieve se utiliza generalmente para generar arañazos virtuales, grietas,

pequeños detalles de la superficie, etc. La profundidad virtual depende del ángulo de la cámara. Cuando el valor se acerca al blanco, el resultado parece que sobresale de la superficie y el negro tiende a empujarlo hacia la superficie. No depende de los vértices.

Normal: este tipo de mapa se utiliza para proyectar detalles del modelo de alta resolución en un modelo de baja resolución. Sobre la base del mapa cromático, una textura de mapa normal cambia la normalidad de un píxel (ver fig. 9).

Displacement: un mapa de desplazamiento depende de los vértices. Este mapa genera profundidad física basada en el valor de escala de grises del mapa.

Ambient occlusion: se utiliza un mapa de oclusión ambiental para generar las sombras o la oscuridad causadas por la cercanía de los objetos (ver fig. 9).

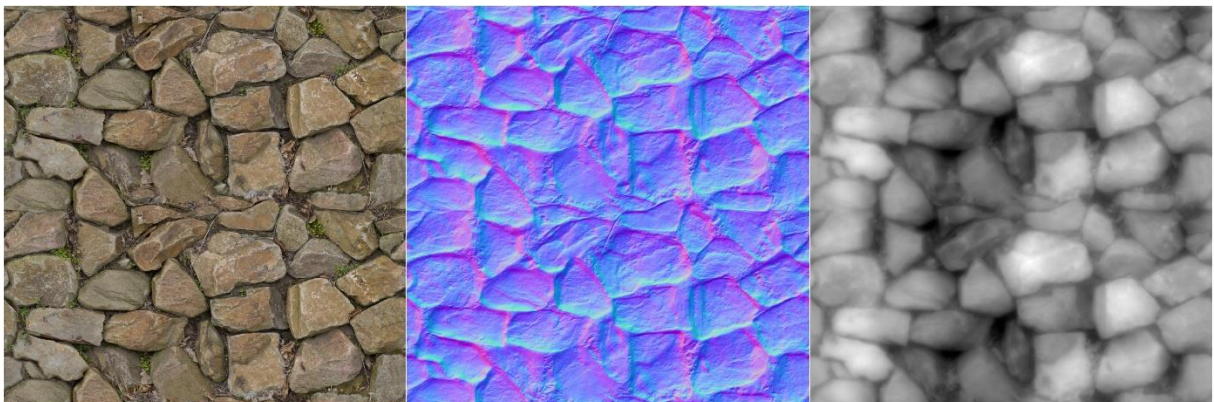


Figura 9. Mapa de diffuse, normal y ambient occlusion respectivamente

Existen algunos tipos más de mapas que son de mucha utilidad en un flujo de trabajo como: *ID maps*, *parallax maps*, *glossiness maps*, *vector displacements*, *height maps*, *fuzz maps*, *roughness maps*, *cavity maps*, *curvature maps*, *translucency or thickness maps*, etc.

2.2.3 Softwares de texturización PBR en tiempo real

Existen muchos softwares en el campo de la texturización 3D en tiempo real, algunos de los más populares son: Mari, Quixel Mixer, Body Paint 3D, ArmorPaint, etc. En el presente trabajo usaremos Substance Painter¹ para el Horneado (bake) y texturización PBR en tiempo real del modelo.

Substance Painter

Substance Painter es una aplicación de pintura y texturizado 3D en tiempo real. Substance Painter permite a los usuarios crear texturas basadas en PBR usando pintura a mano y flujos de trabajo procedurales. Originalmente se desarrolló para su uso en películas y juegos, pero actualmente se usa ampliamente en las industrias de arquitectura, diseño de productos y diseño de automóviles. Esta aplicación se utiliza casi universalmente en la industria de los videojuegos para texturizar objetos 3D. Además, varios grandes estudios cinematográficos también están introduciendo Substance Painter en sus proyectos.

2.3 Matchmoving

La técnica de matchmoving consiste en recopilar información del mundo real en base al material grabado de una escena, ya sea en un set de rodaje o en un entorno exterior, y recrear la cámara empleada con sus características de movimiento en altura y paneo, distancia focal del lente y su posición relativa a elementos y actores, en un espacio tridimensional generado por ordenador para posteriormente poder situar cualquier tipo de elemento 3D dentro de la escena previamente rodada y que este elemento coincida en perspectiva, profundidad, movimientos de cámara en relación con los demás elementos y actores del entorno real, logrando la perfecta integración de la imagen real con la generada por ordenador (Dobbert, 2012). Algunos de los softwares más populares de *matchmoving* son: 3D *Equalizer*, PFtrack, SynthEyes, aunque usaremos Blender ya que posee la capacidad de hacer *matchmoving* en su apartado de efectos visuales.

¹ www.allegorithmic.com

2.3.1 Flujo de trabajo de *Matchmoving*

El propósito de una cámara es recopilar la información tridimensional del mundo real y aplanarlo en 2 dimensiones, por el contrario, en el proceso de *matchmoving* el objetivo es conseguir, mediante la información 2D de una toma, recrear el entorno tridimensional del mundo. Para lograr ejecutar esta tarea el *matchmover*¹ generalmente trabaja con el siguiente flujo de trabajo (Dobbert, 2012).

1. Evaluación del material

La evaluación del material es quizás la parte más importante del flujo de trabajo ya que en esta etapa podremos determinar la complejidad del *matchmoving*, los posibles puntos de *tracking* que emplearemos para poder generar la escena y es aquí donde se deben de plantear algunas preguntas fundamentales como:

- o ¿Qué está haciendo la cámara? ¿Esta estática o se está moviendo? ¿Qué tipo de movimiento está realizando y a qué velocidad?
- o ¿Qué podemos ver en la escena? ¿hay puntos de *tracking*²? ¿algo obstaculiza los puntos de *tracking*?
- o ¿Qué elementos deben ser colocados en la escena? ¿Qué tan prolijo debe ser el *matchmoving*?
- o ¿Quién y cómo va a usar el *matchmoving*?

Existen más preguntas que se pueden plantear, pero estas estarán determinadas por la complejidad de la escena y la experiencia del *matchmover*².

2. Aplicación de información

En esta etapa del flujo de trabajo se recopila la información de la cámara como: la distancia focal, apertura, formato y tipo de filme. Además, se utilizan las mediciones realizadas en el set, como la altura de la cámara, la distancia focal y medidas entre objetos presentes en la escena.

3. Definición de la cámara

Para definir la cámara el *matchmover* utiliza la información recopilada en un software que permita, mediante la fotogrametría, rastrear la cámara, para esto el *software*

¹ Técnico encargado de las tareas de *matchmoving*.

² Los puntos de *tracking* son puntos referenciales del video que el software usara como base para calcular la distancia de la cámara, el movimiento realizado por este, etc.

primero rastrea elementos estáticos dentro de la escena para analizar su movimiento con respecto a la cámara para luego mediante fórmulas matemáticas lograr emular el movimiento de la cámara real y generar la virtual.

4. Montaje del set

Ahora que la cámara está resuelta el *matchmover* debe de generar la escena 3D, dependiendo de los requerimientos del proyecto, para que los posteriores departamentos en el flujo de trabajo puedan trabajar. Esto generalmente se hace con geometrías básicas generadas por el *matchmover* o entregadas por el departamento de modelado, la idea es trabajar con geometría de baja resolución que simplemente sirva de apoyo visual para los animadores.

5. Evaluación del matchmove

En esta etapa el *matchmover* se encarga de probar que el *matchmove* está bien ejecutado, una forma de poder apreciar si hay algún desfase con los movimientos es generar un plano o geometrías con textura de tablero de ajedrez que permita valorar de mejor manera si existe o no algún problema con el *matchmove* (ver fig. 10).

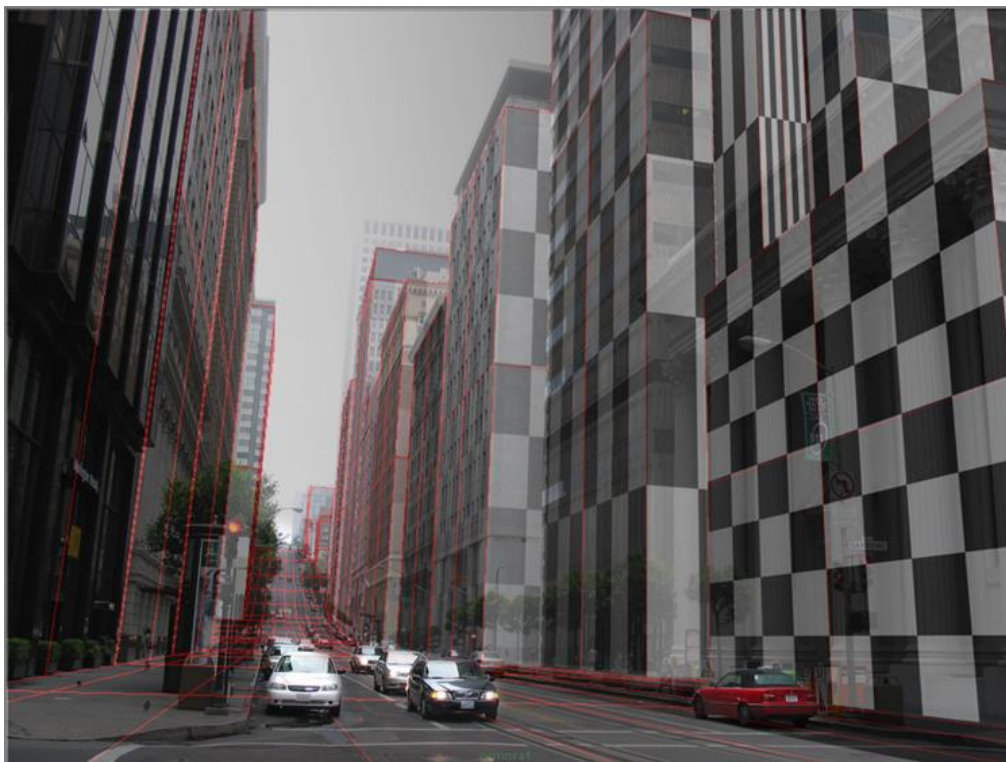


Figura 10. Geometrías con textura de ajedrez (Dobbert, 2012).

6. Entrega de la escena

La entrega de la escena consiste en generar un fichero correctamente organizado con una nomenclatura estandarizada que pueda ser comprendida por los demás artistas, en un formato previamente establecido para su correcta ejecución en los demás departamentos (ver fig. 11).



Figura 11. Archivo con nomenclatura.

2.4 Imágenes HDR (HDRi)

2.4.1 Rango dinámico

“El rango dinámico es el contraste general más alto que se puede encontrar en una imagen” (Bloch, 2013), se refiere a la diferencia de brillo entre las partes más oscuras y brillantes de una escena, por ejemplo, en una escena con bajo rango dinámico habría poca diferencia de brillo entre las partes más iluminadas y menos iluminadas de la escena mientras que con un rango dinámico alto las diferencias de brillo serían de mayor contraste.

2.4.2 ¿Qué es una imagen HDR?

La técnica del *High Dynamic Range* hace referencia al proceso en el cual se toman varias fotos con diferentes tipos de exposición que van desde baja, media y alta exposición, de una misma escena, generalmente con trípode, que capturarán diferentes grados de detalle para posteriormente ser procesadas en un software HDR, que se encarga de unir la información de estas imágenes conservando el mayor grado de información y detalle, el resultado final es conocido como imagen HDR o HDRi (Cooper, 2015).

3. Realización

En este apartado se aborda por completo el proceso por medio del cual se ha ejecutado este proyecto, dividido en las 3 fases de la producción: preproducción, producción y postproducción.

3.1. Preproducción

3.1.1 Referencias

En la etapa de preproducción se inicia la búsqueda de referencias y se define la estética final deseada para el modelo. Para el presente trabajo decidí hacer un tributo al protagonista de la película *The Iron Giant*, película animada de ciencia ficción de 1999, por lo cual ya inicio con un concepto 2D que debo trasladar al 3D (Bird, 1999).

Existe una enorme cantidad de referencias del personaje por lo que tomé un poco de cada estilo, elegí referencias de 3D de alta calidad, 3D para video juegos, ilustración, concept art y miniaturas o figuras de acción (ver fig. 12).

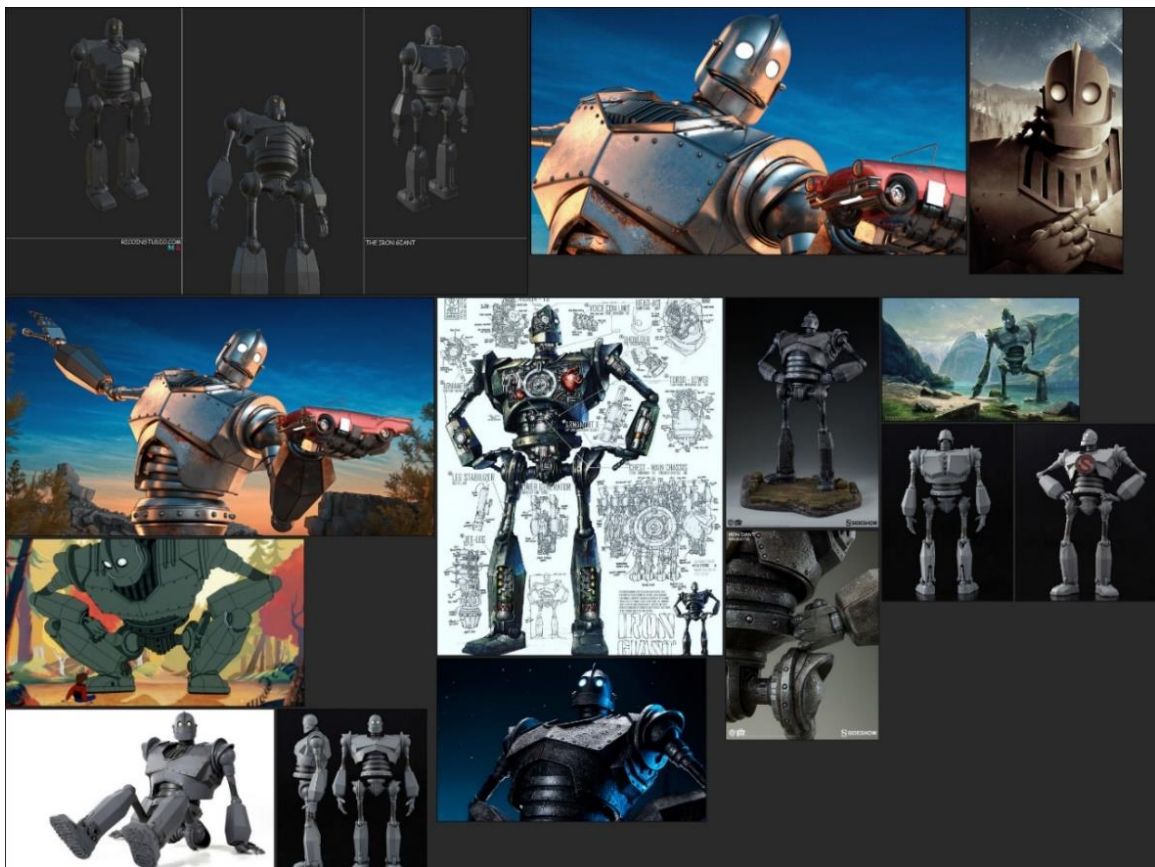


Figura 12. Referencias para diseño de Iron Giant

Decidí utilizar una referencia de bajo poligonaje para empezar a modelar el *blockout*¹ del personaje, también busqué referencias que sirvan de guía para las vistas frontal, lateral y posterior dentro del software, utilicé referencias de mayor grado de realismo para empezar a visualizar los detalles más específicos a modelar y poder preparar una estrategia adecuada a las necesidades del objeto. Por último, adicione referencias para la posterior etapa de texturización para tener una idea de la calidad de las texturas deseadas.

3.2. Producción

3.2.1 Modelado de *Iron Giant*

En la etapa inicial del modelado se aconseja empezar con un *blockout* del modelo que servirá de guía para entender de mejor manera las proporciones y relaciones de tamaños entre las formas a modelar. El *blockout* no es un modelo detallado ni preciso, es simplemente una herramienta que nos permitirá tener un entendimiento global del modelo, anticipar posibles problemas y así trazar la mejor estrategia para su desarrollo en el menor tiempo posible.

Una vez que se determinan las formas básicas y técnicas a emplear en el modelo se puede avanzar a la realización del *High poly*, en mi caso me decidí empezar por un modelado poligonal de las piernas del robot. El modelado poligonal se suele empezar con la ayuda de una forma básica o primitiva lo más cercana a las formas que requerimos y con una topología de baja densidad para la facilidad de edición, este puede ser un cubo, esfera, plano, cono, cilindro, etc (ver fig. 13).

¹ Modelo base que define tamaños y proporciones básicas.

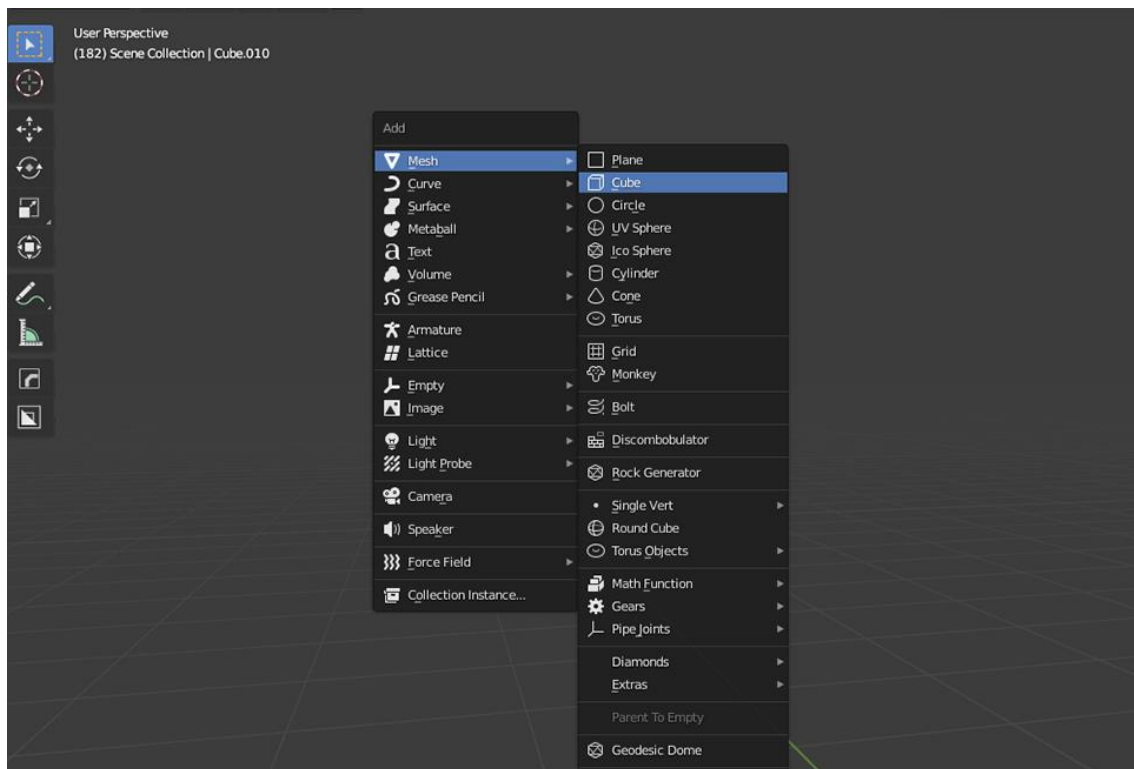


Figura 13. Insertando formas básicas en el modo objeto.

Dentro de Blender existen un conjunto de modos con diferentes propiedades y funciones los cuales se utilizan dependiendo de lo que se desee editar ("Blender 2.91 Manual de Referencia — Blender Manual", 2020) (ver fig. 14).

- *Object mode*: El modo predeterminado, disponible para todos los tipos de objetos, ya que está dedicado a la edición de bloques de datos de objetos (por ejemplo, posición, rotación, tamaño).
- *Edit mode*: Un modo disponible para todos los tipos de objetos renderizables, ya que está dedicado a su edición de bloques de datos de objeto de "forma" (por ejemplo, vértices / bordes / caras para mallas, puntos de control para curvas / superficies, trazos / puntos para lápiz grueso, etc.).
- *Sculpt mode*: Un modo solo de malla, que habilita la herramienta de escultura 3D de malla de Blender.

- *Vertex Paint mode*: Un modo solo de malla, que le permite establecer los colores de los vértices de su malla (es decir, "pintarlos").
- *Weight Paint mode*: Un modo solo de malla, dedicado a la ponderación de grupos de vértices.
- *Texture Paint mode*: Un modo solo de malla, que le permite pintar la textura de su malla directamente en el modelo, en las Vistas 3D.
- *Particle edit mode*: Un modo solo de malla, dedicado a los sistemas de partículas, útil con los sistemas editables (cabello).
- *Pose mode*: Un modo solo de armadura, dedicado a la presentación de armadura.
- *Draw mode*: Un modo solo de lápiz grueso, dedicado a crear trazos de lápiz grueso.










Icon	Name
	Object Mode
	Edit Mode
	Sculpt Mode
	Vertex Paint Mode
	Weight Paint Mode
	Texture Paint Mode
	Particle Edit Mode
	Pose Mode
	Draw Mode

Figura 14. Iconos de modos en Blender.

Para la edición del cubo se activa el *edit mode* donde se puede empezar a editar las propiedades de los vértices, aristas y polígonos del cubo, dentro del *edit mode* se selecciona el icono de *vertex select* y con la herramienta de selección se selecciona los vértices individuales para su edición en las coordenadas X, Y, Z. siguiendo las imágenes de referencias de las vistas frontal y lateral (ver fig. 15).

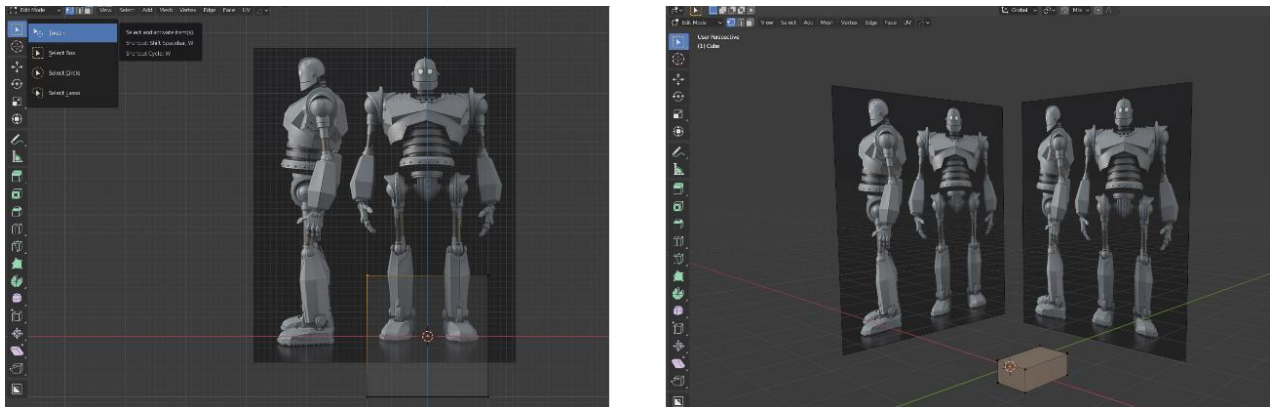


Figura 15. Edición de malla en el edit mode.

Además de la edición de los vértices individuales en posición, escala o rotación existen otras formas de editar la malla, como agregar líneas de corte, líneas en bucle, extrusión de caras, etc. Con la sumatoria de estas herramientas se logra editar la malla a la forma deseada (ver fig. 16).

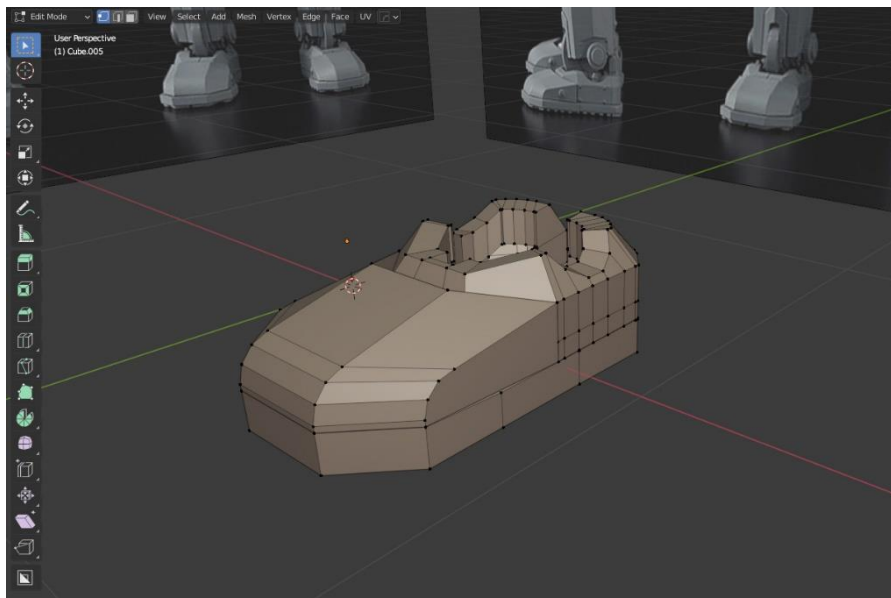


Figura 16. Modelado de pie.

El proceso se repite con cada forma del modelo hasta tener toda la estructura de la pierna modelada y con la ayuda del modificador de simetría se genera de forma automática la otra pierna sin la necesidad de volver a modelar, además se agrega el modificador de subdivisión que mediante una operación procedural divide las caras de la malla dándole mayor suavidad a está generando una impresión de mayor definición y calidad sin la necesidad de agregar densidad real a la misma optimizando en recursos (ver fig. 17).

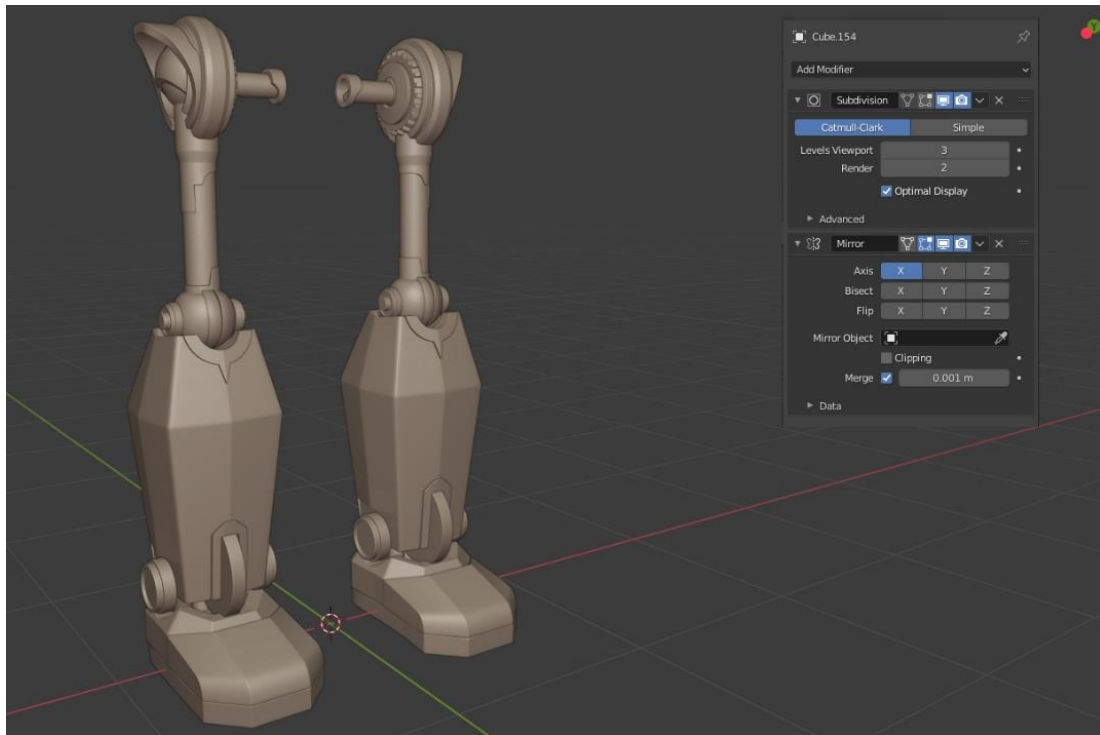


Figura 17. Modelado completo de extremidades inferiores.

Se repite el mismo proceso, de agregar forma básica de bajo poligonaje en el *object mode* y su modificación en el *edit mode*, con las demás partes del Iron Giant hasta completar el *High poly* (ver fig. 18).

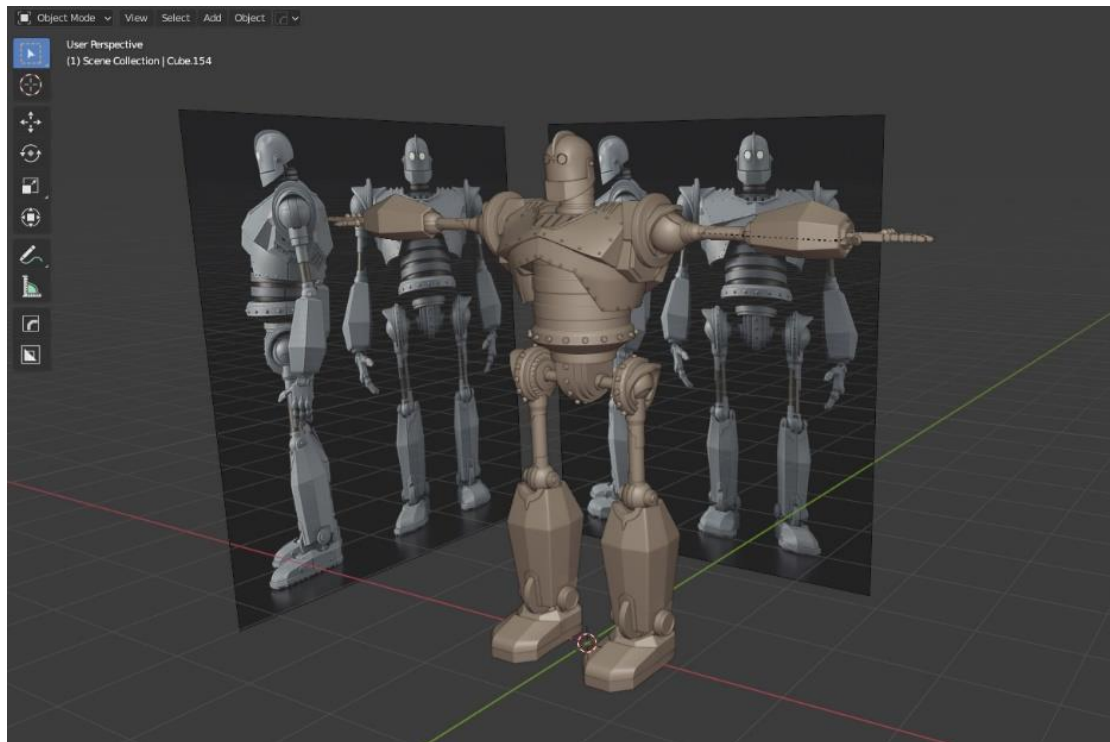


Figura 18. Modelado completo del Highpoly.

Una vez concluido el modelado del *High poly* se puede avanzar al *low poly*, que consiste en reducir la cantidad de polígonos de la malla para facilitar, por ejemplo, el despliegue de mapas uv para texturización o la deformación de malla en animación y *rigging*. En el caso del presente trabajo el *low poly* servirá para el despliegue de mapas uv y para el *bake* del *High poly* sobre el *low poly* creando la percepción de que el *low poly* tiene tantos polígonos como el *high poly*, pero sin ser tan pesado.

Para el proceso de creación del *low poly* se duplica el *high poly* y se le quita el modificador de subdivisión, el modificador de *mirror* se deja intacto hasta llegar a la etapa del despliegue de mapas uv, una vez realizados estos ajustes se procede a limpiar la malla quitando el exceso de aristas que se emplearon para la creación de biseles en el *high poly*, es importante que esta reducción de polígonos no afecte la forma básica del modelo logrado en el *high poly* debido a que para el correcto *bake* del *high poly* sobre el *low poly* las diferencias de distancias entre los polígonos del *high* y *low poly* deben ser mínimas de lo contrario pueden surgir problemas en la posterior etapa de texturización (ver fig. 19).

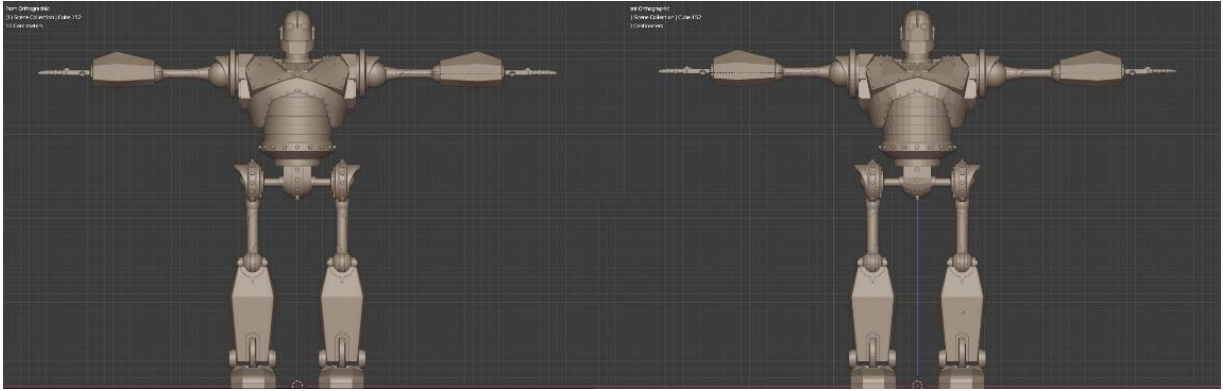


Figura 19. High y Low poly respectivamente.

3.2.2 Mapeado UV

El mapeado UV es el proceso en el cual se crea una imagen de 2 dimensiones a partir de un objeto de 3 dimensiones, por ejemplo, imaginemos una caja de cereales que consta de 3 dimensiones, al ser cortada de forma estratégica, puede ser desplegada y colocada sobre una mesa representando solo 2 dimensiones (ver fig. 20).

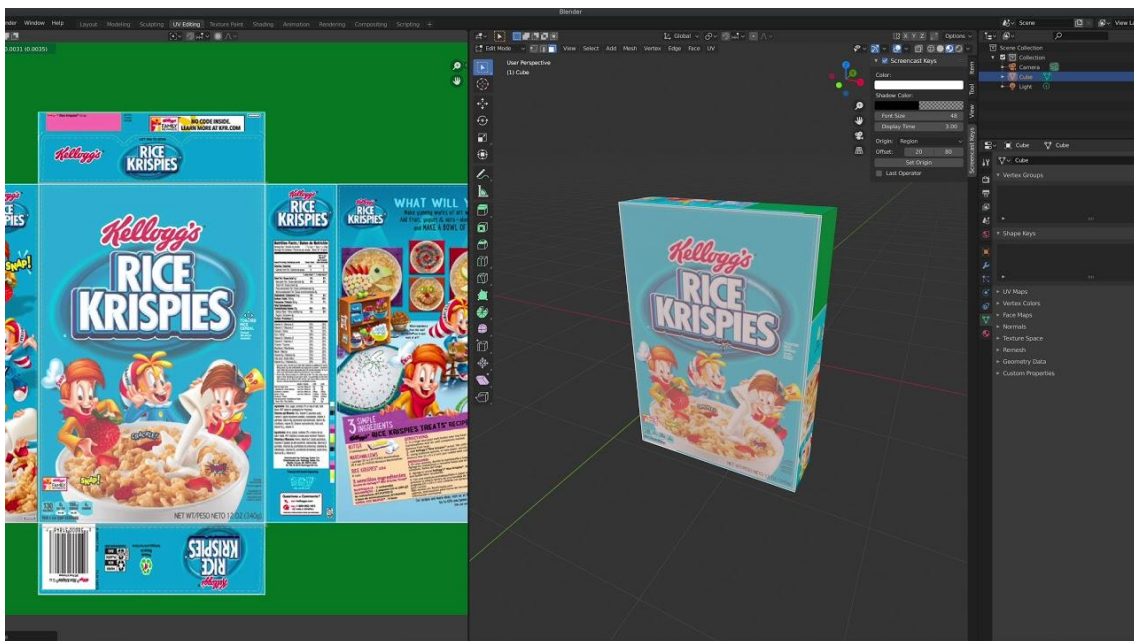


Figura 20. Ejemplo de caja de cereal, a la izquierda el entorno 2D y a la derecha el entorno 3D.

Para el despliegue de mapas UV dentro de Blender es necesario acceder al espacio de trabajo de UV *editing* en el cual encontraremos un diseño predeterminado del espacio de trabajo dividido en dos, un área para el *uv editor* y otro para el *viewport* (ver fig. 21).

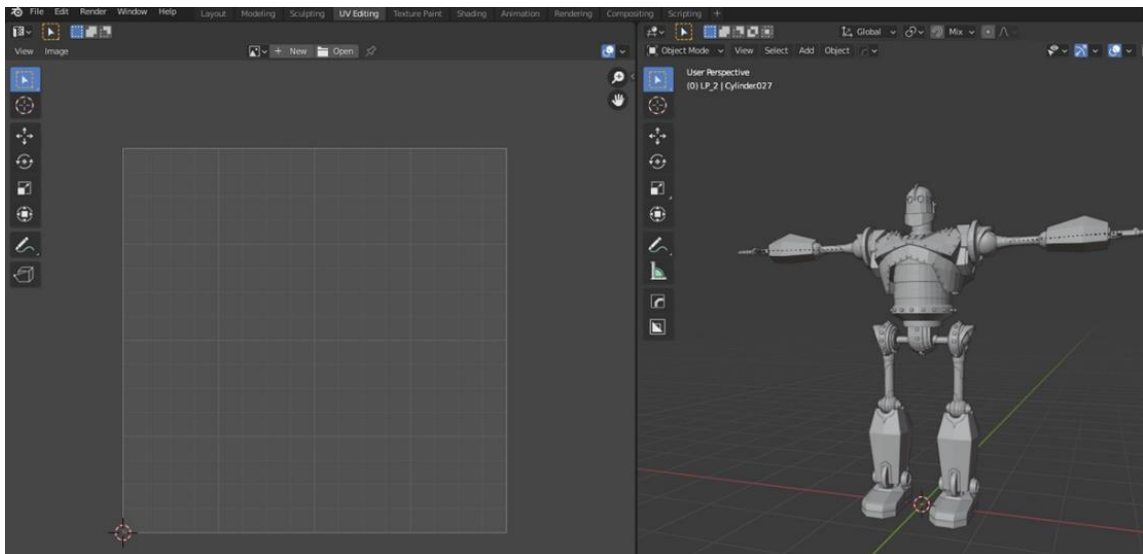


Figura 21. Espacio de trabajo UV editing.

El despliegue de mapas UV consiste en convertir la superficie del espacio 3D en una superficie 2D que nos permita luego texturizar con mayor facilidad, este proceso comienza al generar las líneas de corte que facilitarían el despliegue de la malla sobre las coordenadas UV, estas líneas de corte deben ser manejadas con precaución en base a los requerimientos del proyecto ya que estos cortes pueden ser notorios al momento de texturizar si no se los ubica adecuadamente, por ejemplo, en un personaje orgánico se suele colocar las líneas de corte por debajo de los brazos que son áreas menos visibles para el espectador (ver fig. 22).



Figura 22. UV map de modelo orgánico del libro *Digital Modeling* de Willian Vaughan.

En el espacio de trabajo del UV *editing* se accede al área del *viewport* donde se activa el *edit mode* del objeto, dentro del *edit mode* se activan las herramientas para la selección de las aristas y líneas de corte (ver fig. 23).

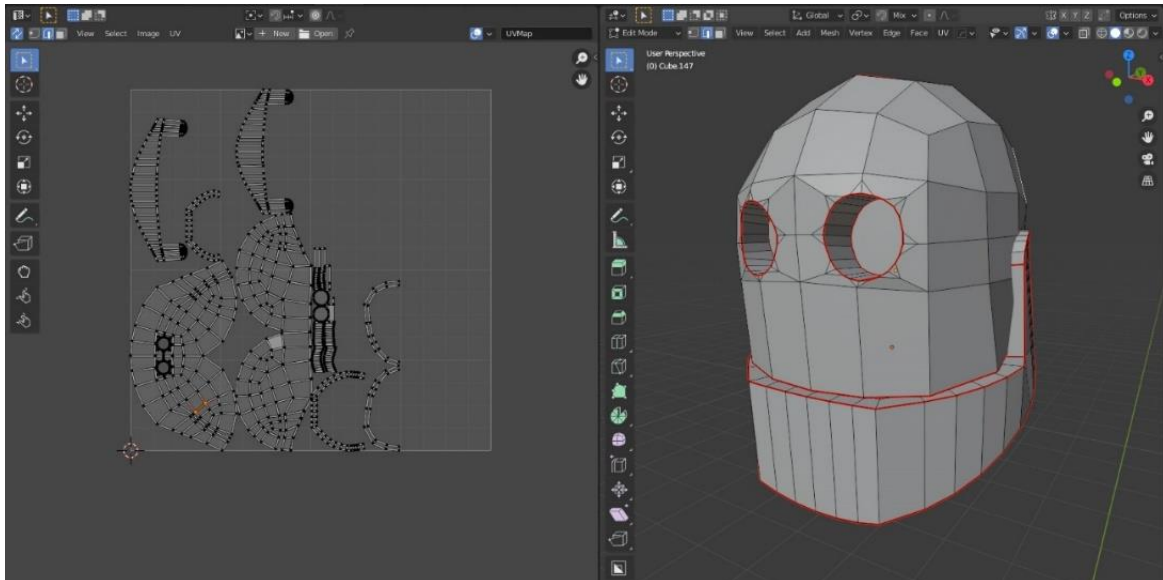


Figura 23. Mapa desplegado en el UV editor, líneas de corte marcados con rojo en el viewport.

Una vez que las líneas de corte han sido generadas correctamente se procede a desplegar la malla sobre las coordenadas UV, esto se hace en el *viewport* seleccionando la totalidad de los polígonos de la malla para luego en el UV *editor* dar *click* derecho y seleccionar la opción de *unwrap* para desplegar los mapas de forma automática, después de que los mapas están desplegados se comienza a optimizar el área utilizada para que los mapas tengan la mayor resolución posible, ahora que los mapas están correctamente desplegado se aplican los modificadores de *mirror* de los objetos, como brazos y piernas, para reordenarlos en el mapa UV, de esta manera se ahorra la tarea de desplegar mapas UV en, por ejemplo, brazo derecho e izquierdo por separado (ver fig. 24).

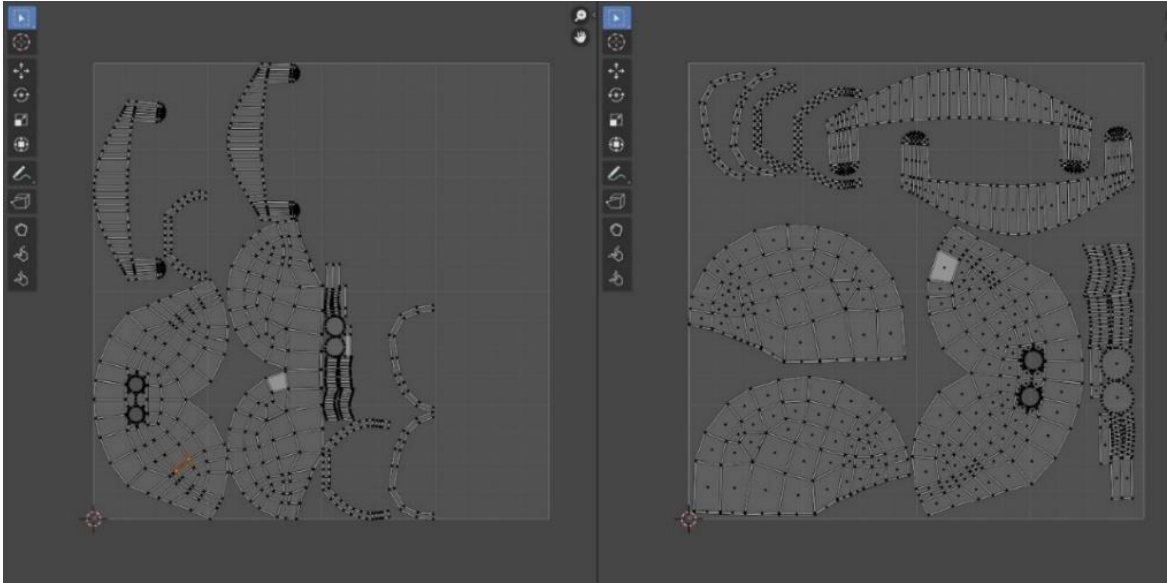


Figura 24. A la izquierda mapa desplegado automáticamente, a la derecha mapa reorganizado por toda el área.

Siendo que el objetivo de este proyecto es que el modelo 3D tenga una apariencia real y convincente es necesario que sus texturas tengan mucha resolución, debido a esto decidí utilizar 8 mosaicos de mapas UV ganando mayor resolución para las texturas, un mosaico para cada extremidad del robot, 1 para la cabeza y los 3 restantes para el tronco del robot (ver fig. 25).

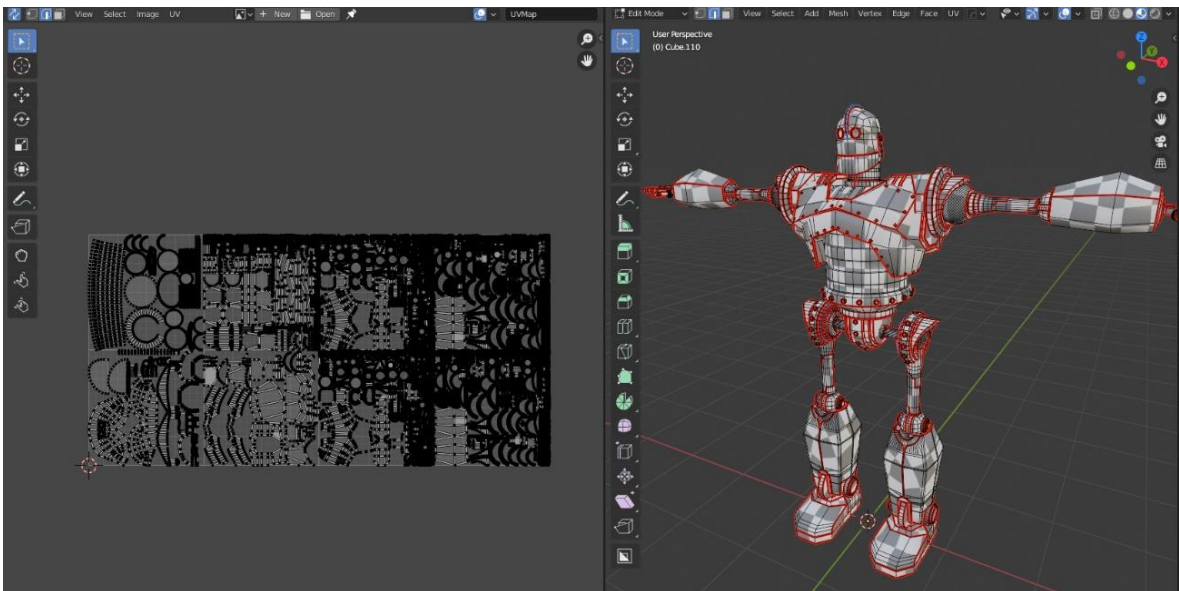


Figura 25. UDIMs, mapas desplegados en 8 mosaicos de texturas.

3.2.3 Texturizado en Substance painter

Una vez que la malla ha sido modelada y sus mapas UV estén correctamente desplegados, se puede proceder a la texturización PBR dentro del entorno de Substance painter.

3.2.3.1 Importar malla a Substance painter

Antes de poder texturizar el modelo se debe exportar la malla *high* y *low poly* en formato .obj¹ o .fbx² desde blender con todos su modificadores aplicados, para el *low poly* previamente se debe de suavizar sus sombras en el modo objeto (click derecho *shade smoth*) (ver fig. 26).

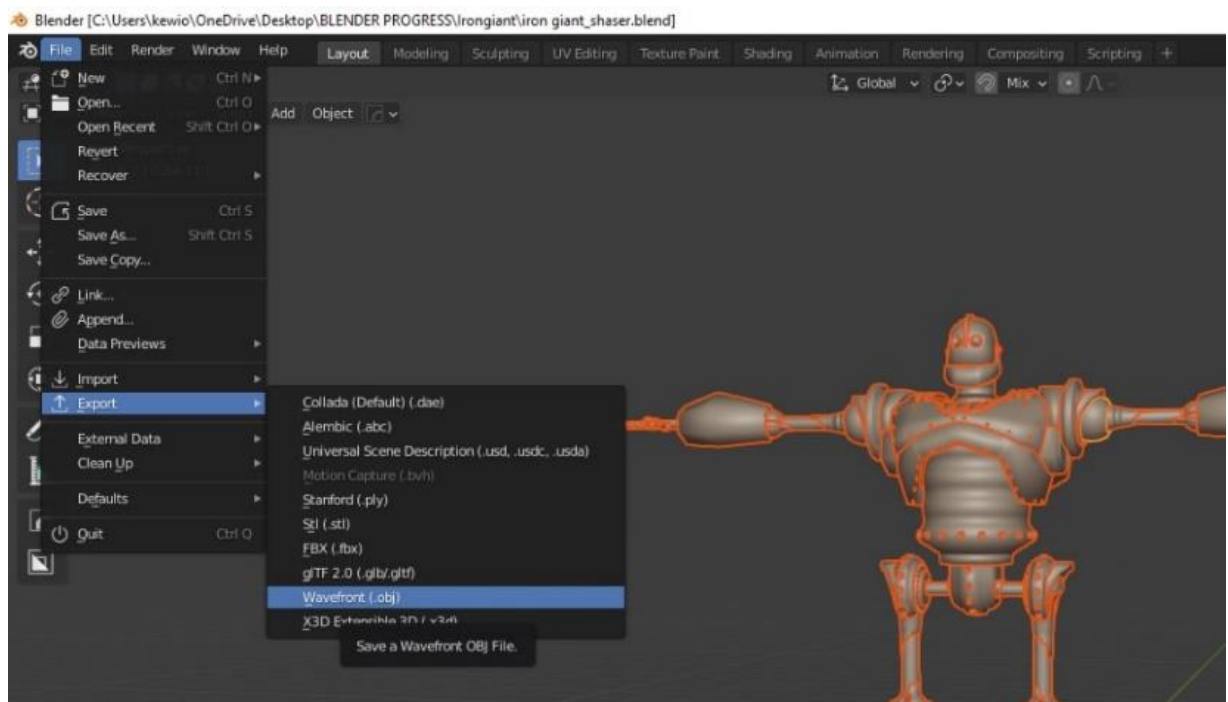


Figura 26. Exportación de malla low poly.

¹ Extensión de archivo 3D estándar que contiene información del objeto 3D, coordenadas, mapas de texturas.

² Formato de archivo que se encarga de mantener todos los elementos, funcionalidad e información del archivo original, para trabajar de manera compartida y simultánea entre varias aplicaciones DE diferentes desarrolladores.

Dentro de *substance painter* se procede a configurar el proyecto y a importa la malla *low poly* del *Iron Giant* (ver fig. 28)., para el proyecto se selecciona un *template* PBR – *Metallic Roughnes*¹, una resolución del documento de 2048 y un formato de *normal map* *OpenGL*², que es el que utiliza por defecto Blender (ver fig. 27).

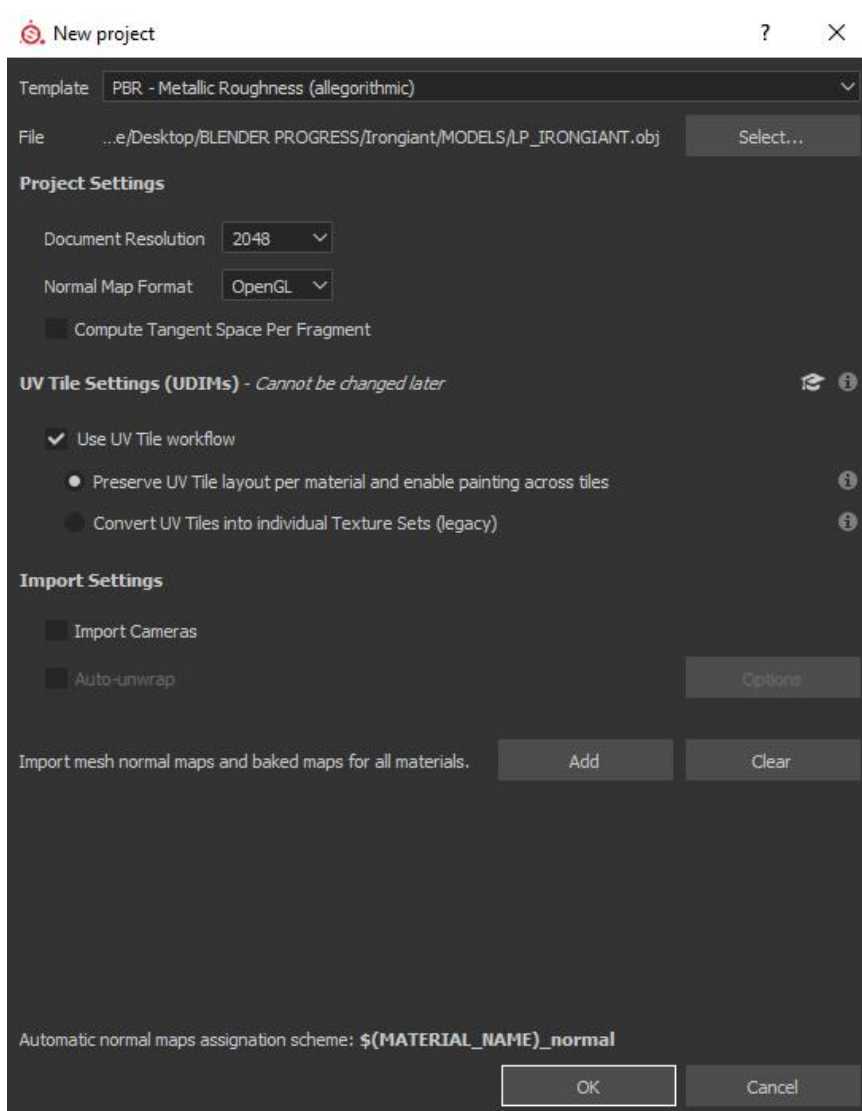


Figura 27. Configuración de proyecto en *substance painter*.

¹ Plantilla PBR de *substance painter* que pose canales de: base color, *height*, *roughness*, *metallic* y *normal*.

² Formato de *normal map* que nos dice que el canal rojo apunta a la derecha, el azul a la dirección de la normal y el verde apunta hacia arriba, ideal para trabajar en Blender.

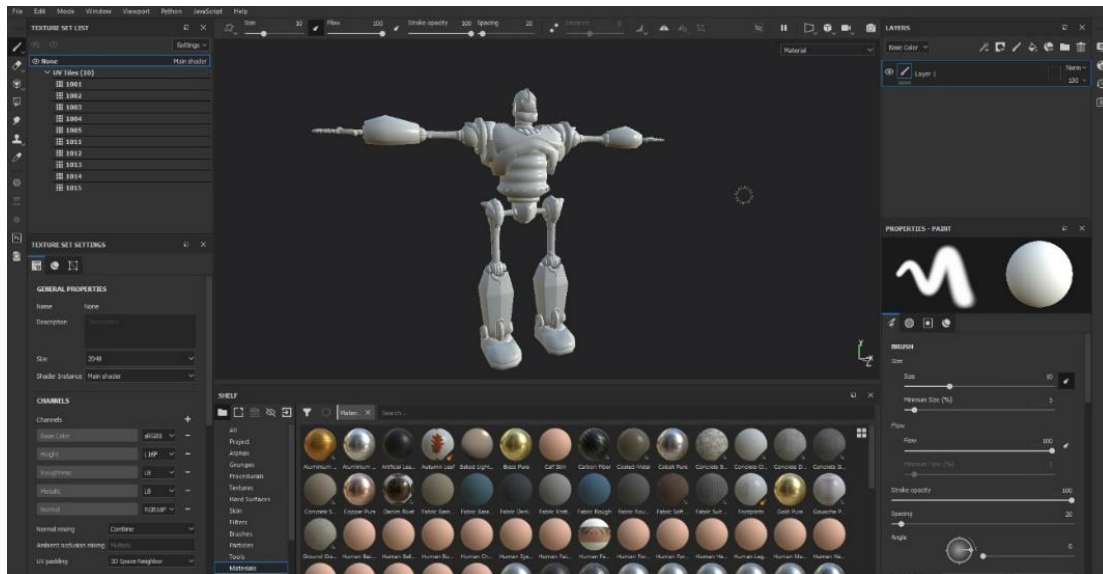


Figura 28. Modelo low poly importado en substance painter

3.2.3.2 Bake de high poly sobre low poly

El *bake*¹ hace referencia a la transferencia de información de la malla sobre una textura para después ser utilizados por los *shaders*² y *smart materials*³ para efectos más avanzados de texturizado. Para el *bake* se accede al visor del *texture set setting* y en el apartado del *mesh map* se da click en *bake mesh maps* para abrir la ventana de *Baking*. Dentro de la ventana de *Baking* seleccionamos los mapas de *normal*, *world space normal*, *ambient occlusion*, *curvature*, *thickness* y *position*. Se configura un tamaño de salida de 2048, por último, se trae la malla de *high poly* de donde *substance* obtendrá la información que necesita para el correcto *bake* (ver fig. 29).

¹ Horneado.

² Pieza de código que se ejecuta en la GPU que sirve para determinar cómo una superficie en particular responde a la luz, simular profundidad de campo, resplandores o brillos, etc.

³ Tipo de material de creación procedural que se adapta a las formas e imperfecciones de la malla 3D.

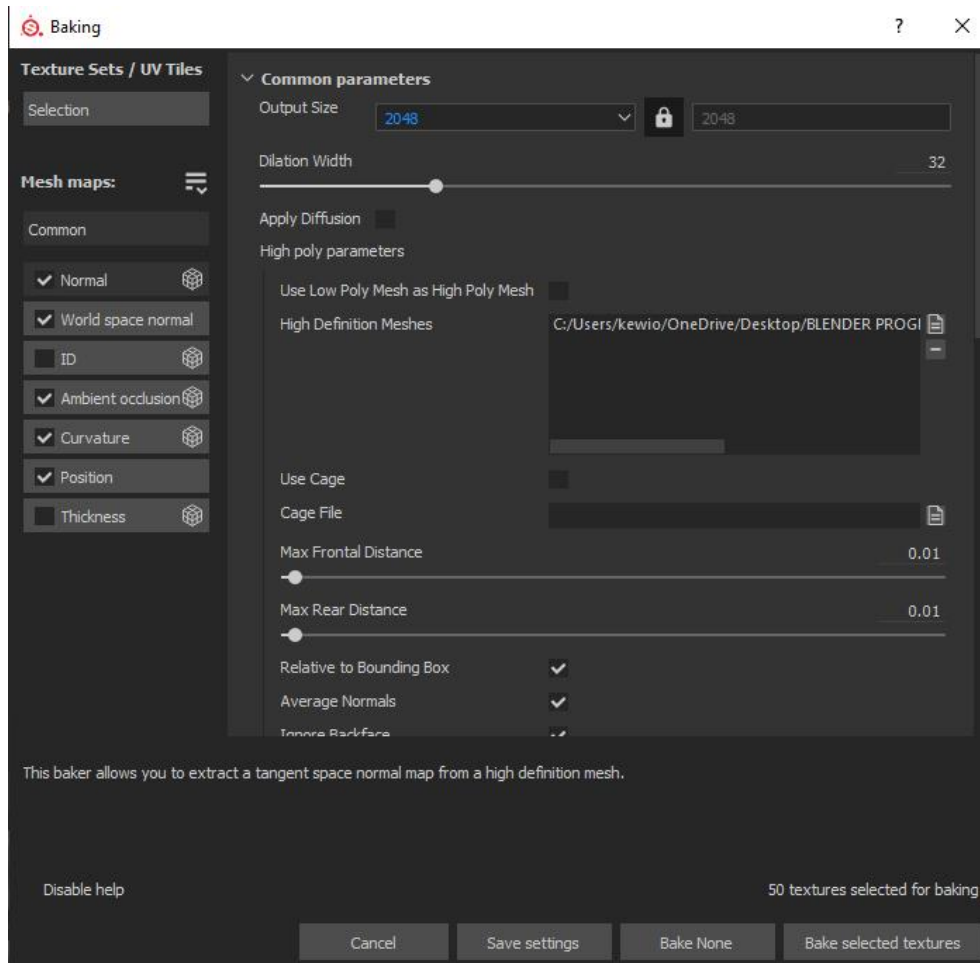


Figura 29. Ventana de Baking

El proceso de *bake* puede tardar varios minutos y lo aconsejable es dejar al ordenador para que pueda utilizar todos sus recursos en esta tarea.

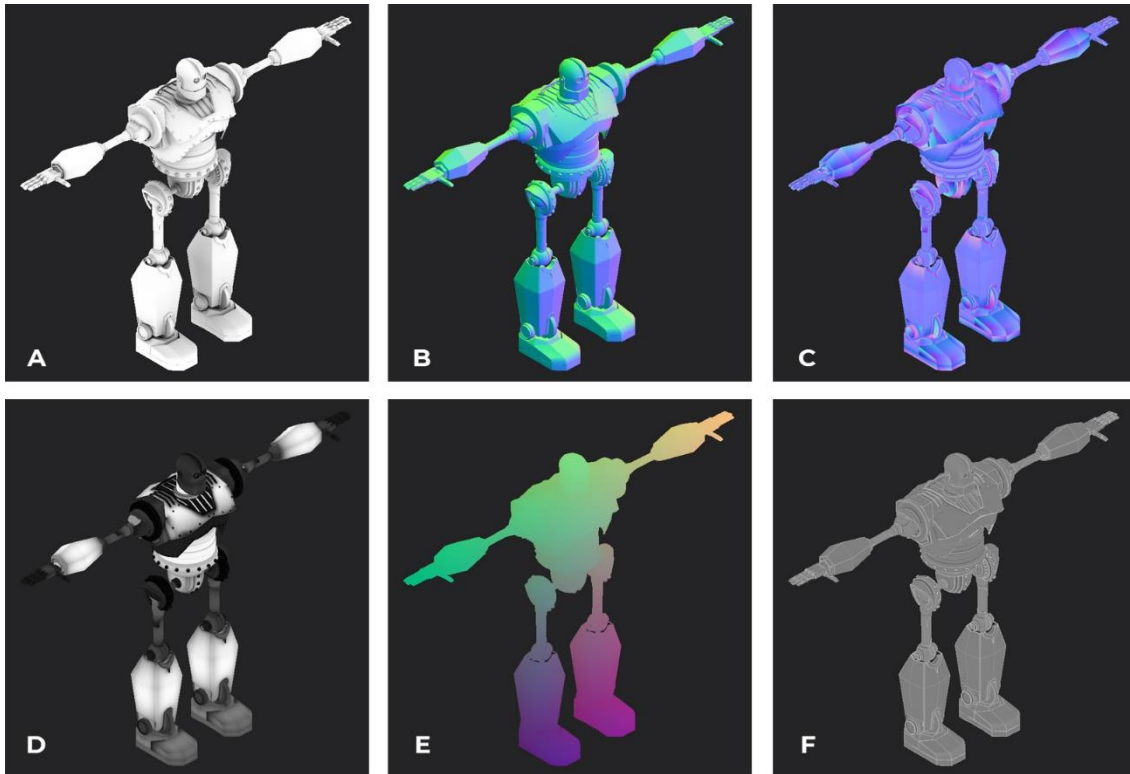


Figura 30. Mapas de texturas resultantes del bake

El resultado del *bake* nos proporciona 6 tipos de mapas: (ver fig. 30).

- A. *Ambient occlusion*: genera sombras provocadas por la cercanía de los objetos.
- B. *World space normal*: tipo de *normal map* que almacena las coordenadas normales de un objeto con respecto a la dirección del espacio mundial.
- C. *Normal*: se utiliza para falsificar información de iluminación de detalles de alta resolución en una malla de baja resolución sin tener que agregar más geometría.
- D. *Thickness*: mapa en escala de grises que define el grosor de las diferentes áreas de una malla. Las áreas más oscuras representan las partes más delgadas de una malla de modelo, mientras que el blanco representa las partes más gruesas de un modelo.
- E. *Position*: almacena las coordenadas 3D de cada punto de una malla en su respectivo píxel de textura en el UV. Esto permite al usuario aplicar de forma procedural¹ efectos que dependen de la posición.

¹ Creado mediante un algoritmo

F. *Curvature*: mapa en escala de grises que almacena y representa las áreas cóncavas y convexas de una malla. Este mapa se utiliza generalmente para generar información procedural, como daños y desgaste.

Con estos mapas *substance painter* tiene la información necesaria para poder utilizar sus algoritmos y lograr así una texturización procedural, por medio de sus *smart materials*, ahorrando la tarea de pintar manualmente sobre el modelo.

3.2.3.3 Aplicando *smart materials* y texturas.

Ahora que el modelo ha pasado por el proceso de *bake* de forma satisfactoria se puede empezar a aplicar *smart materials*, para eso hay que ir a la ventana del *shelf*, luego se selecciona el apartado de *smart materials* en donde se encuentran las texturas procedurales, en esta parte del flujo de trabajo el objetivo es encontrar una textura base que se acerque lo más posible al resultado final deseado (ver fig. 31).

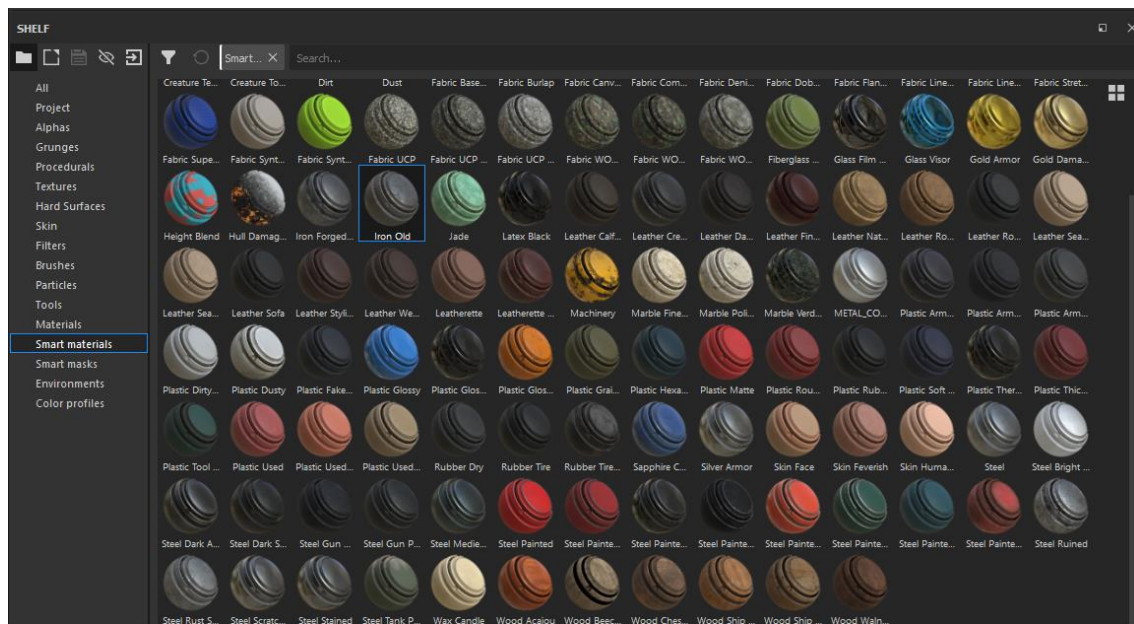


Figura 31. *Smart materials* en la ventana de *shelf*

El *Smart material* más cercano a lo buscado es un *Iron Old*, que muestra propiedades metálicas semejantes a hierro, signos de desgaste, envejecimiento del metal y una sutil capa de óxido (ver fig. 31).



Figura 32. Smart material base aplicado sobre modelo 3D

En las referencias encontradas el metal tiene un ligero toque azulado, para conseguirlo se agrega un *fill layer*, en el cual solo se le dejara activa la propiedad de color, luego en la opción de *base color* se elige un tono azul. Para que el color base no sea regular en toda la superficie del objeto, y muestre el efecto de desgaste, se agrega un *Alpha* que oculte ciertas partes del *fill layer*, para lograrlo hay que ir dentro de la ventana de *layers* y seleccionar el icono de *add mask*, en donde se desplegará un submenú, y seleccionar *add black mask*, esto ocultará en su totalidad el tono azul debido a que el color negro oculta y el blanco muestra, luego, dentro de la misma ventana de *layers* se selecciona el icono de *add effect*, del cual se desplegaran varios efectos, y elegimos el efecto *add fill*, con este efecto aplicado se procede a seleccionar una máscara que se adecue a lo deseado, en este caso un *grunge map 008*, para finalizar, se baja la opacidad del tono azul a un 15% y se agrega a la capa un modo de fusión de color (ver fig. 33).

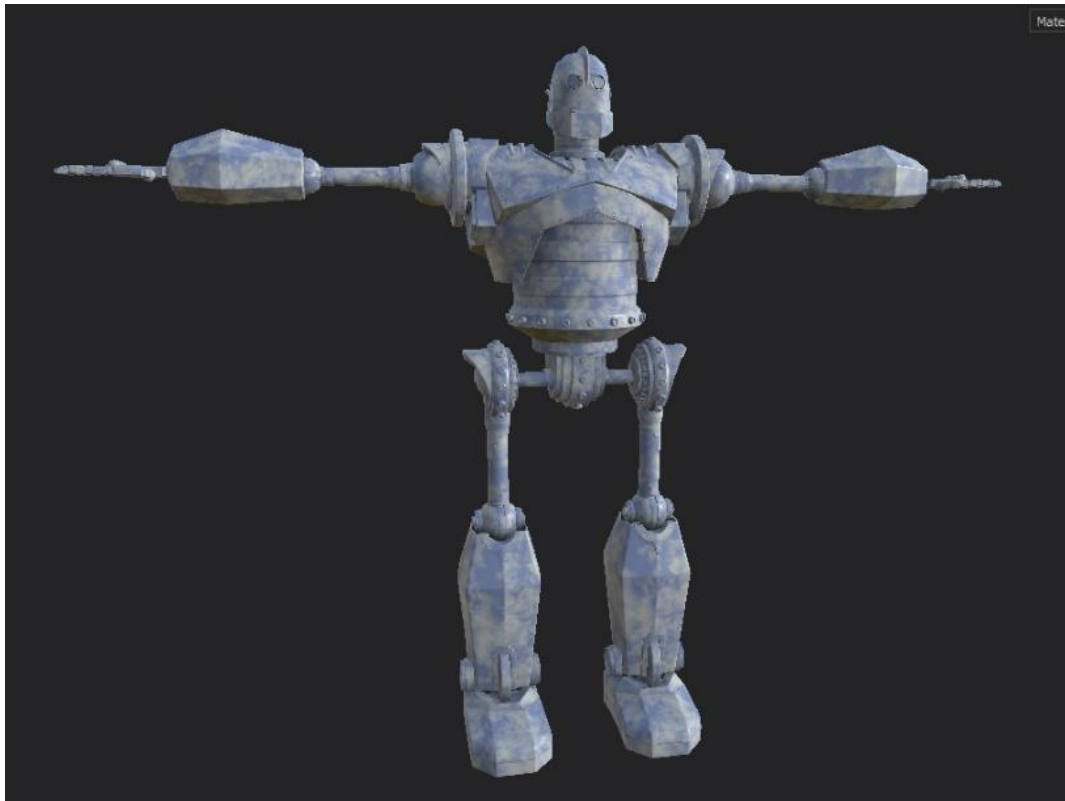


Figura 33. Muestra de máscara grunge aplicado sobre modelo 3D sin el Smart material aplicado.

Para los ojos es necesario agregar un canal extra de *emissive*, que permite genera luz, este se añade desde el *texture set settings* en la sección de canales dando *click* en el signo “+” y seleccionando *emissive* (ver fig. 34).

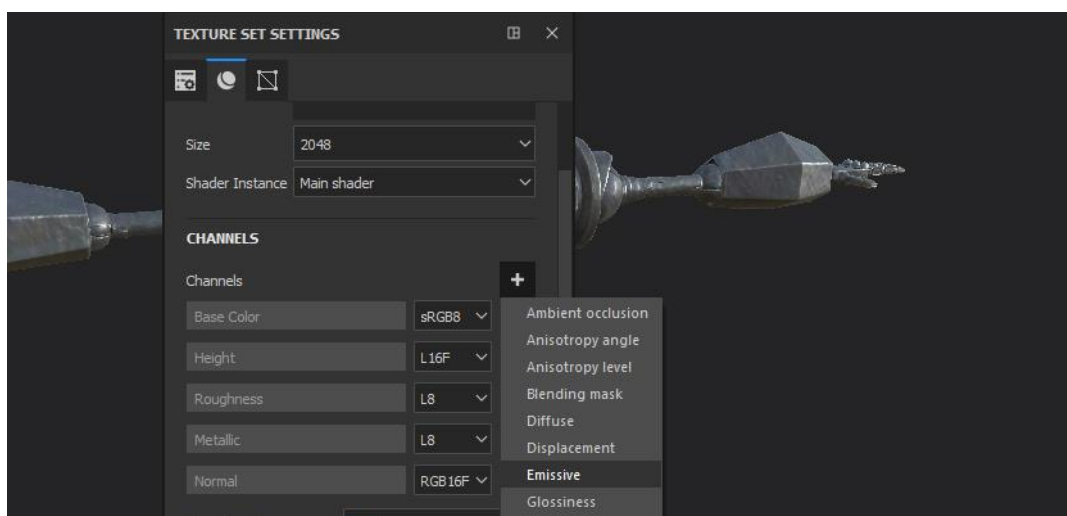


Figura 34. Agregando canal de *emissive* desde el *texture set settings*

Una vez agregado el canal de *emissive* se procede a agregar un *fill layer* en el cual se activará únicamente el *emissive*, en la sección de material del *properties – fill*, con una tonalidad amarilla muy clara (ver fig. 35).

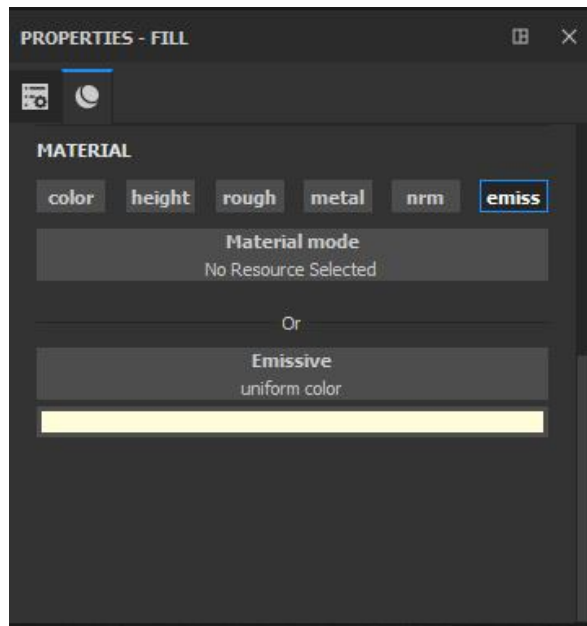


Figura 35. Ventana del Properties - Fill, sección de material.

Posteriormente se procede a agregar, al *fill layer* con el material de *emissive*, una máscara negra, para ocultar el material, y un efecto de *add Paint* (ver fig. 36).

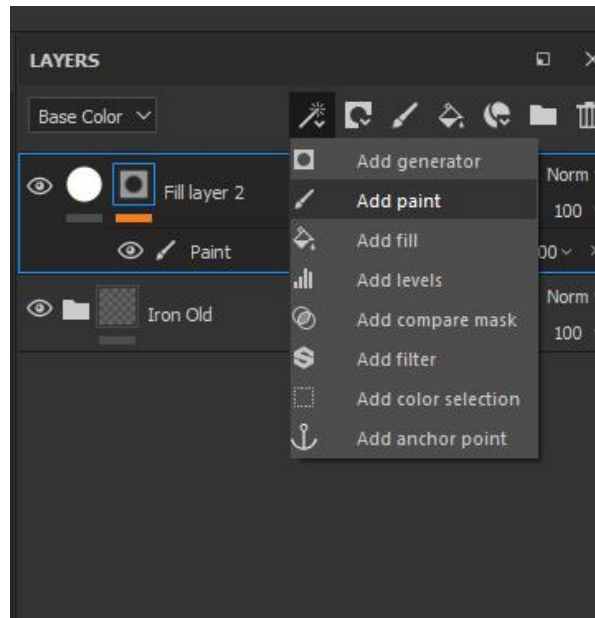


Figura 36. Agregando máscara y efecto de add paint.

Posteriormente, con la máscara seleccionada, activamos la herramienta de *Polygon fill* de la barra lateral izquierda de herramientas y en la ventana de *properties – polygon fill* se selecciona el modo de relleno *polygon fill*, que sirve para hacer una selección poligonal de la malla (ver fig. 37).

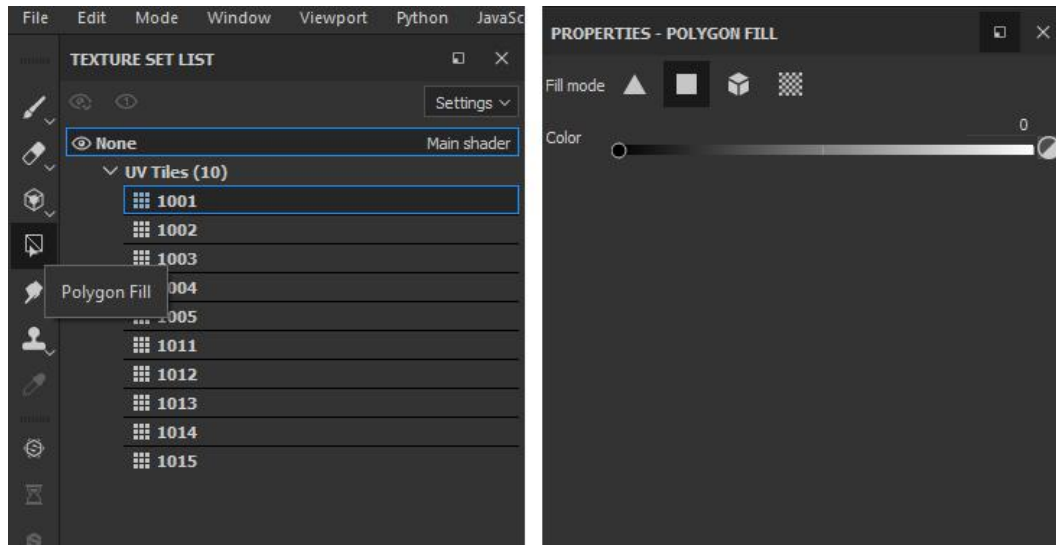


Figura 37. A la izquierda, herramienta *polygon fill*, al otro extremo, ventana de *properties - polygon fill*.

Para seleccionar los polígonos de los ojos, se coloca el color del *polygon fill* en blanco y se procede a seleccionar los polígonos en el *viewport* (ver fig. 38).

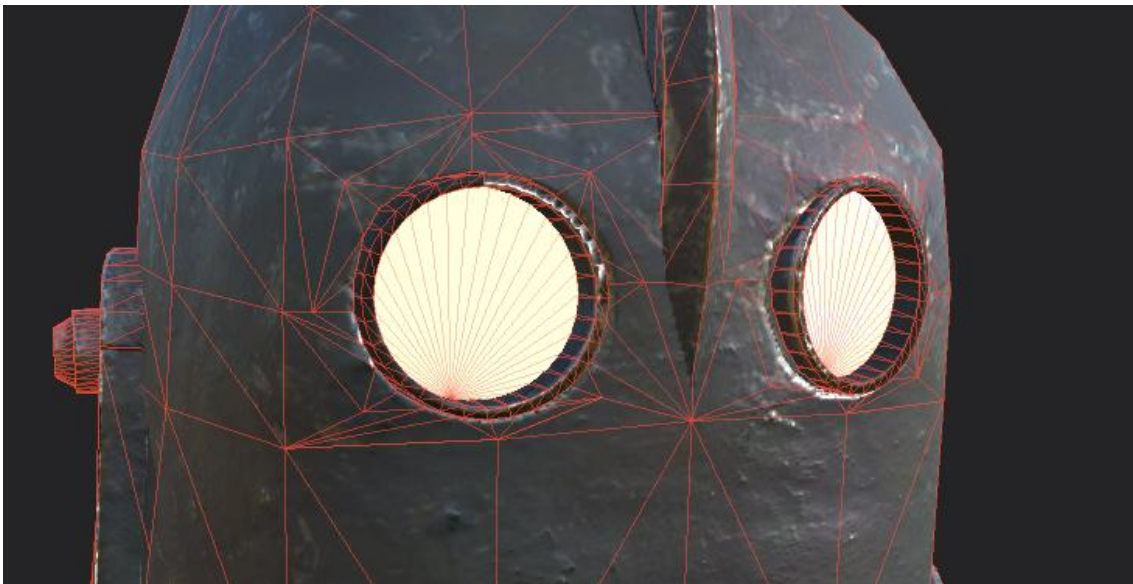


Figura 38. Selección de ojos con herramienta *polygon fill*.

3.2.3.4 Exportando texturas.

Ahora que las texturas están terminadas solo queda exportarlas para llevarlas a Blender, para eso hay que ir a la barra superior de menú y desplegar el apartado de *File* y seleccionar la opción de *Export textures*, el cual abrirá la ventana de dialogo de *Export textura* (ver fig. 39).

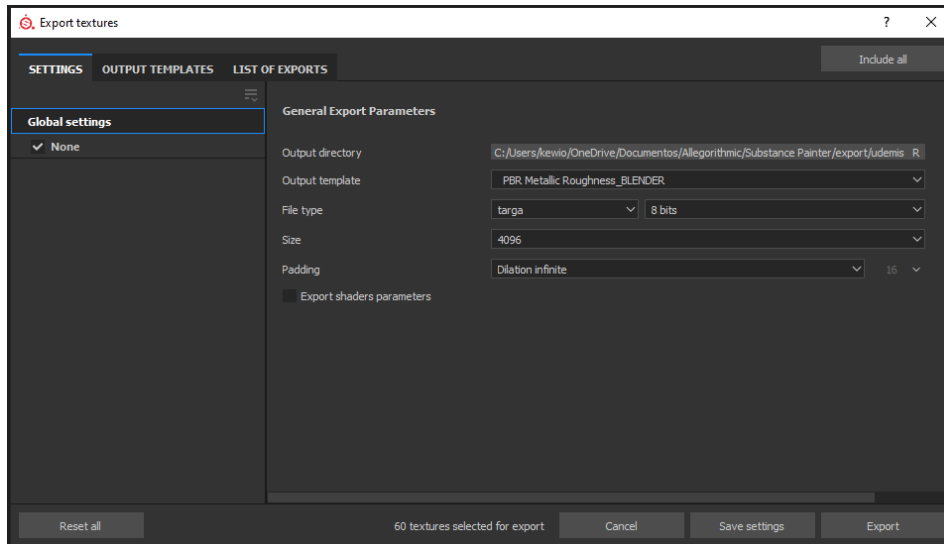


Figura 39. Ventana de dialogo de export texture con paramteros configurados.

En esta ventana se configuran los parámetros de exportación y ruta de las texturas, lo único que quedaría antes de exporta es modificar la plantilla del PBR *Metallic Roughness*, en la pestaña de *Output templates*, para que sus normales sean *OpenGL*, compatible con Blender, ya que por defecto vienen en *DirectX¹* (ver fig. 40).

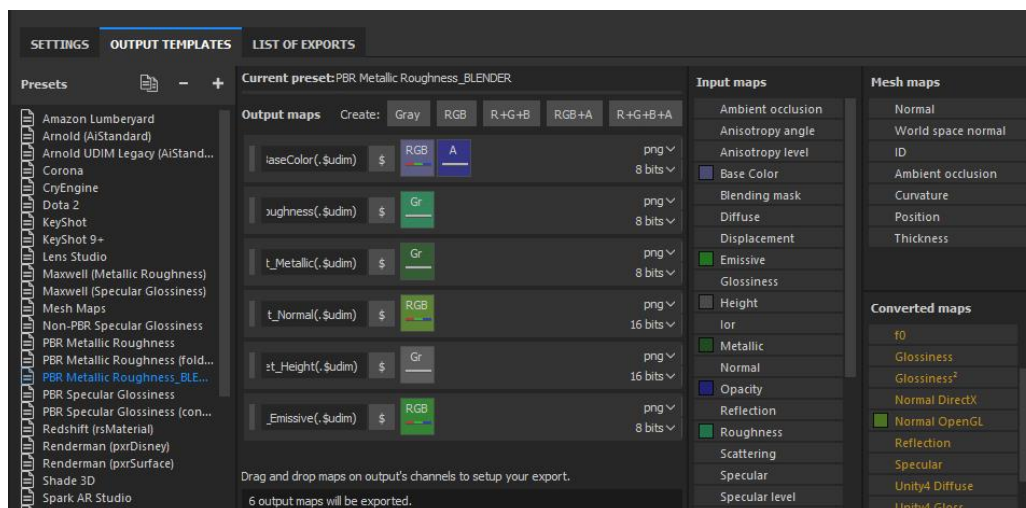


Figura 40. Edición de plantilla de normal diretX a openGL.

¹ Formato de *normal map* que nos dice que el canal rojo apunta a la derecha, el azul a la dirección de la normal y el verde apunta hacia abajo, ideal para trabajar en, por ejemplo, Maya y 3D studio max.

3.2.3.5 Importando texturas a Blender.

Para poder apreciar el resultado final de las texturas exportadas es necesario importarlas al modelo dentro de Blender y hacer una prueba, para eso hay que entrar al espacio de trabajo de *shading* en Blender y seleccionando el modelo ir a *shader editor* donde se procederá a trabajar con nodos para importar las texturas, usaremos un nodo de *image texture* por cada tipo de material que importaremos a la malla, estos nodos se añaden con el comando rápido *shift + A*, seleccionando *texture* y por último *image texture* (ver fig. 41).

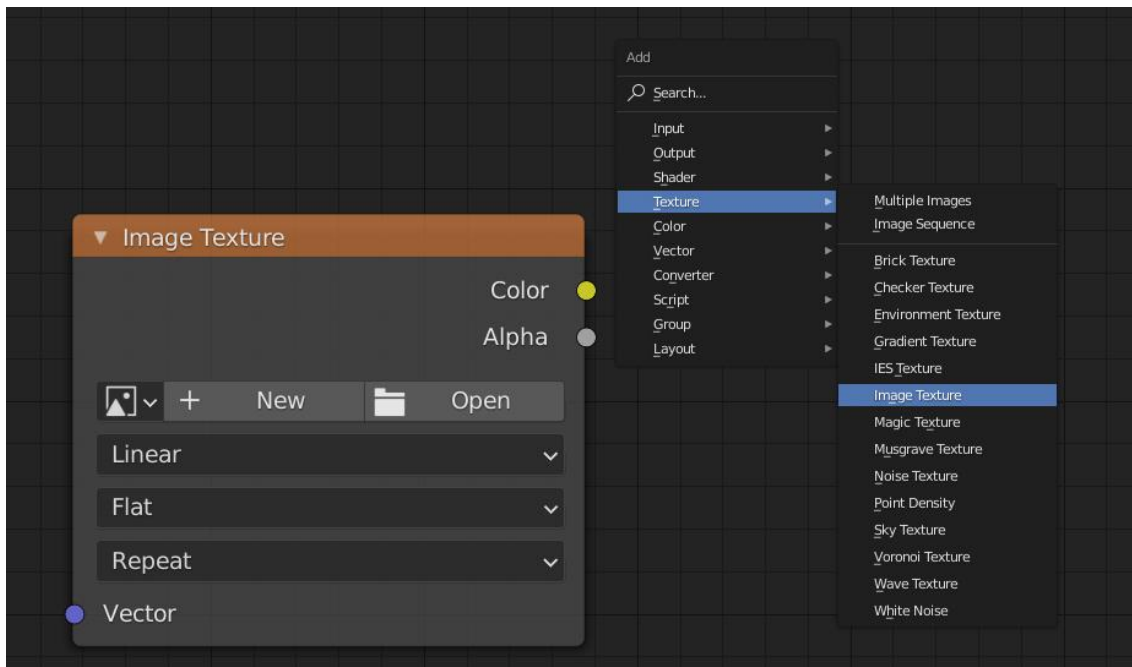


Figura 41. Nodo de image texture.

Una vez añadidos los diferentes tipos de texturas a los nodos de *image textura* solo hace falta conectarlos al nodo principal de *Principled BSDF* con su respectivo tipo de textura con la particularidad de que al nodo de normales se le agrega un nodo de vector *Normal map*, al roughness un nodo de color de *Bright/contrast* y el nodo de *height* se conecta directamente al nodo de salida del material a través de un nodo de *vector displacement* (ver fig. 42).

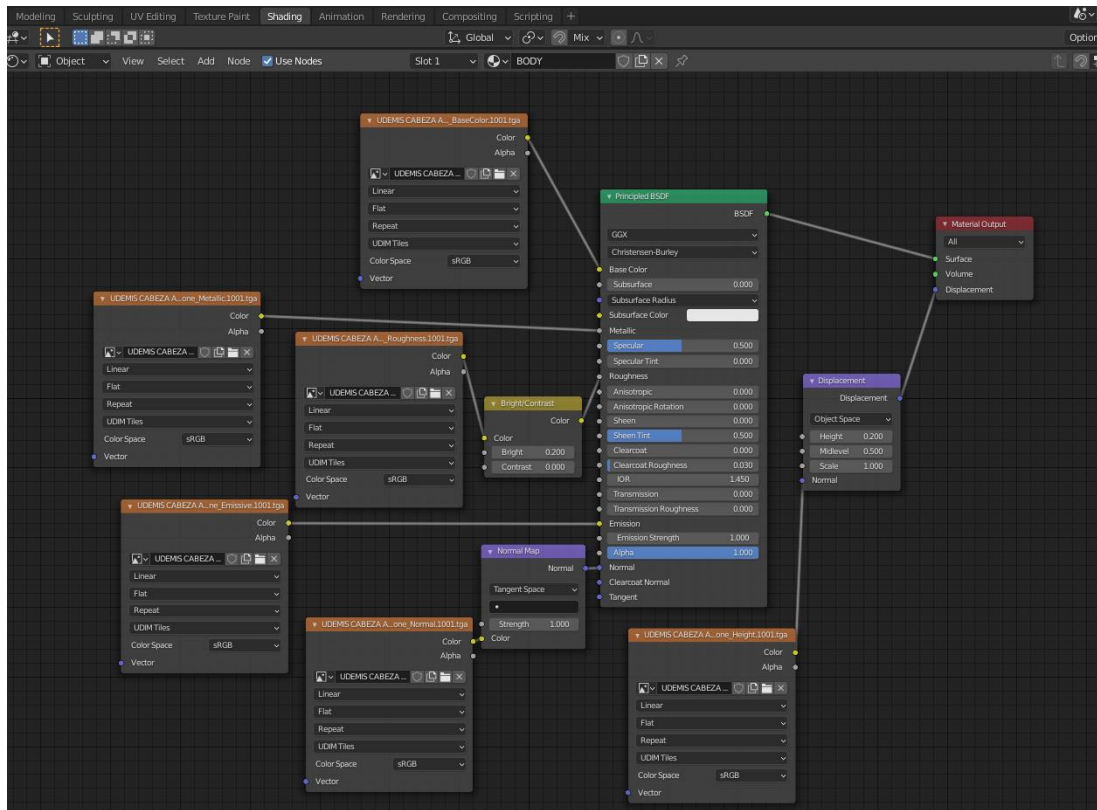


Figura 42. Sistema de nodos de texturas del Iron Giant.

Con las texturas agregadas a el modelo en Blender se puede hacer una prueba rápida de render para determinar si el resultado es el óptimo (ver fig. 43).



Figura 43. Prueba de render PBR dentro de Blender.

3.2.4 *Matchmoving* de material rodado

Para este proyecto se rodó un pequeño video a las afueras del *RCDE stadium*¹ con una cámara OSMO DJI GO². En este apartado se explicará el proceso por medio del cual se logra realizar un correcto *matchmoving* a partir del material de video generado.

3.2.4.1 Evaluación de material rodado

En esta etapa del *matchmoving* nos limitamos a analizar el material rodado, identificar posibles puntos de guía para el seguimiento de movimiento y a responder una serie de preguntas que nos debemos plantear para entender con mayor profundidad lo que esta sucediendo en la escena.

- ¿Qué está haciendo la cámara? ¿Esta estática o se está moviendo? ¿Qué tipo de movimiento está realizando y a qué velocidad?
 - o La cámara está haciendo un paneo hacia la derecha con cámara en mano a una velocidad lenta haciendo un ligero movimiento hacia adelante mientras rota hacia el centro de la toma.
- ¿Qué podemos ver en la escena? ¿hay puntos de *tracking*? ¿algo obstaculiza los puntos de tracking?
 - o Se pueden apreciar 5 posibles puntos de *tracking* en el suelo, otro punto al fondo por los contenedores de basura, uno más al fondo en unas vallas metálicas y uno más a la derecha sobre un símbolo de información, ninguno de estos posibles puntos de *tracking* se pierde durante el movimiento.
- ¿Qué elementos deben ser colocados en la escena? ¿Qué tan prolijo debe ser el *matchmoving*?
 - o Se debe colocar un modelo 3D de *Iron Giant*, debe ser un *matchmoving* de buena calidad ya que el modelo estará en el centro de la toma y el suelo consta de un patrón cuadrículado que delataría un movimiento inadecuado.

Con estas preguntas resueltas se puede empezar a formular una estrategia para la resolución del *matchmoving*.

3.2.4.2 Aplicación de información

En esta etapa se procede a recopilar toda la información que ayude al *software* a calcular el seguimiento de cámara.

- Altura del sensor: 6.16
- Distancia focal: 22 mm
- *Frame rate* del vídeo: 25 Fps
- *Aspect ratio*: 1
- Tipo de movimiento de cámara: Posición y rotación
- Trípode: no
- Altura de cámara: 1.70 m

3.2.4.3 Definición de cámara

Para definir la cámara entramos al espacio de trabajo de *motion tracking* de Blender y agregamos el video que vamos a usar de base para el *matchmoving* (ver fig. 44).

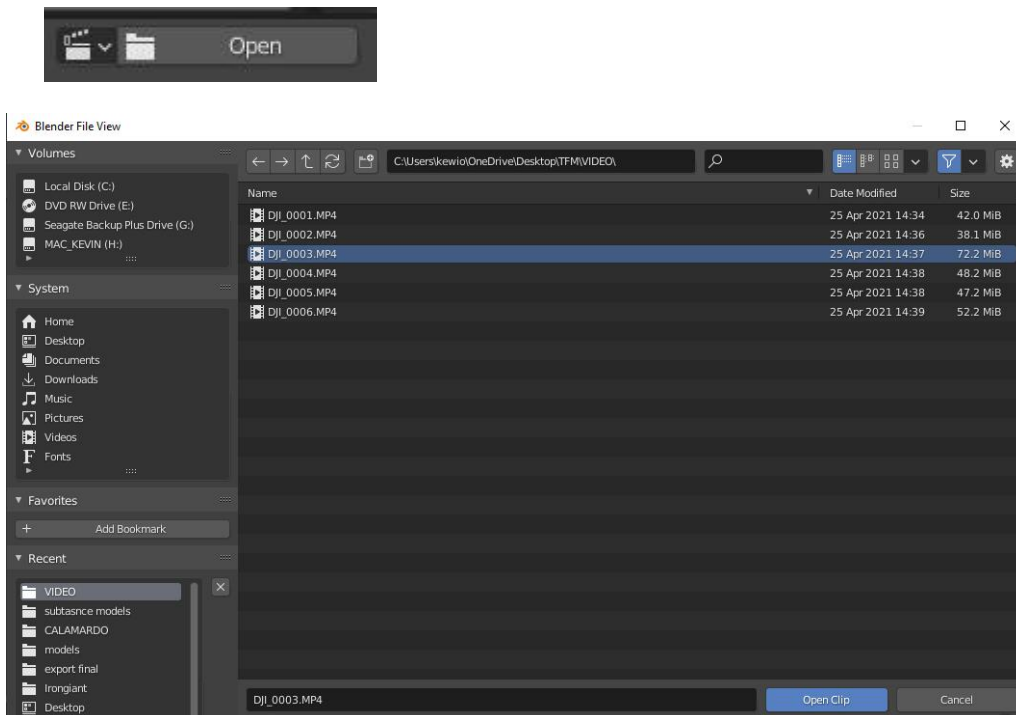


Figura 44. Importación de video a espacio de trabajo de motion tracking.

Una vez que el vídeo esta importado se debe de llenar la información que se ha recopilado de la cámara en los parámetros de cámara en el *software* (ver fig. 45).

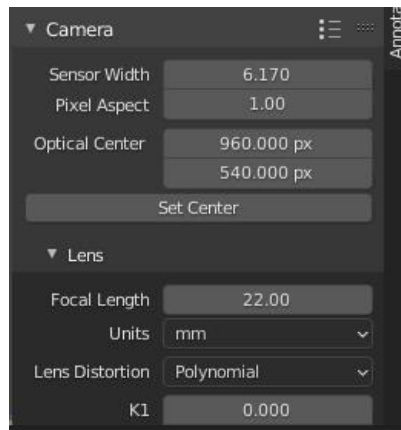


Figura 45. Configuración de parámetros de cámara.

Ahora que la configuración de cámara esta definida continuamos marcando los puntos de *tracking* (ver fig. 46).

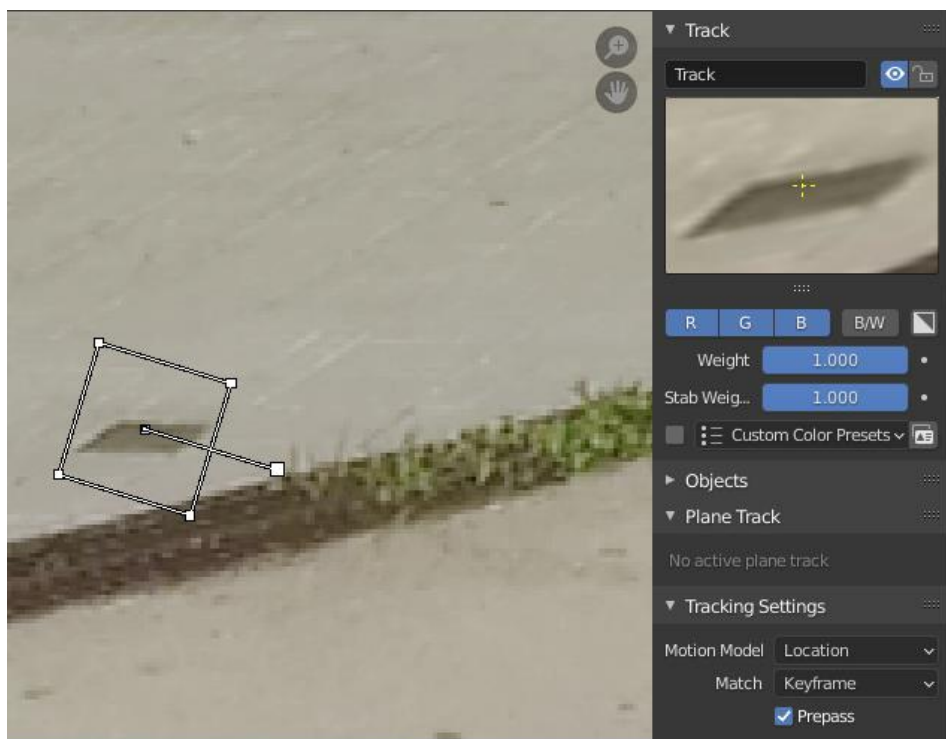


Figura 46. Punto de tracking.

Por defecto Blender nos pide un mínimo de 8 puntos de *tracking* para poder generar los cálculos para la cámara virtual, en este caso se aplicaron 9 puntos y al ser resueltos

dieron una tasa de error del 0.36 %, lo que significa que el *software* logró resolver satisfactoriamente la cámara virtual ya que lo recomendable es tener una tasa de error inferior al 1 % (ver fig. 47).

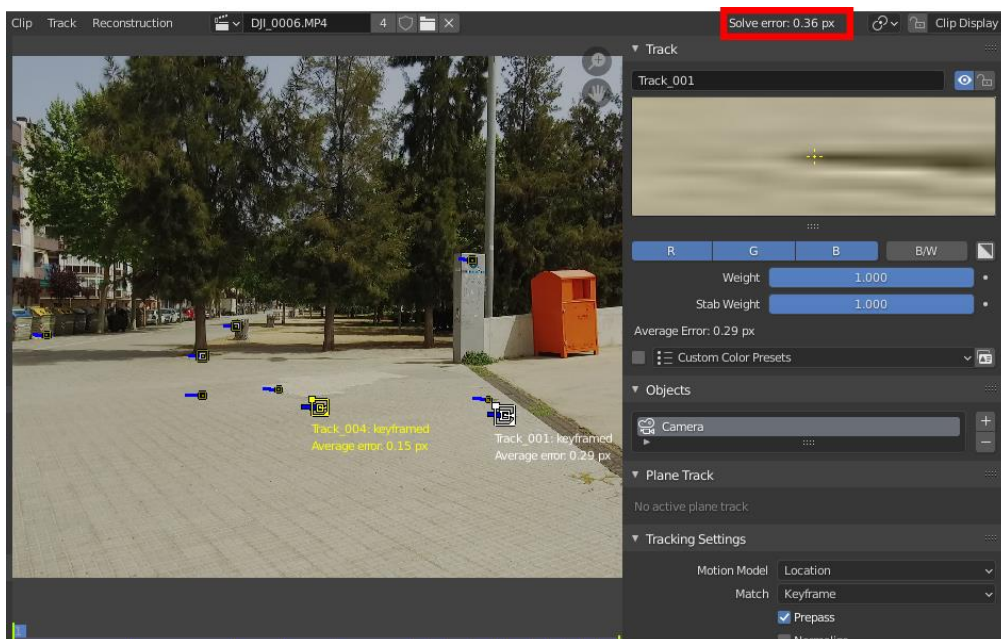


Figura 47. Puntos de tracking resueltos con una tasa de error del 0.36 %

3.2.4.4 Montaje del set

Con la cámara virtual resuelta es hora de empezar a añadir el set virtual, esto implica, en el caso de este proyecto, agregar un suelo base sobre el que se situará el modelo 3D y el modelo 3D del *Iron Giant* en *low poly* (ver fig. 48).

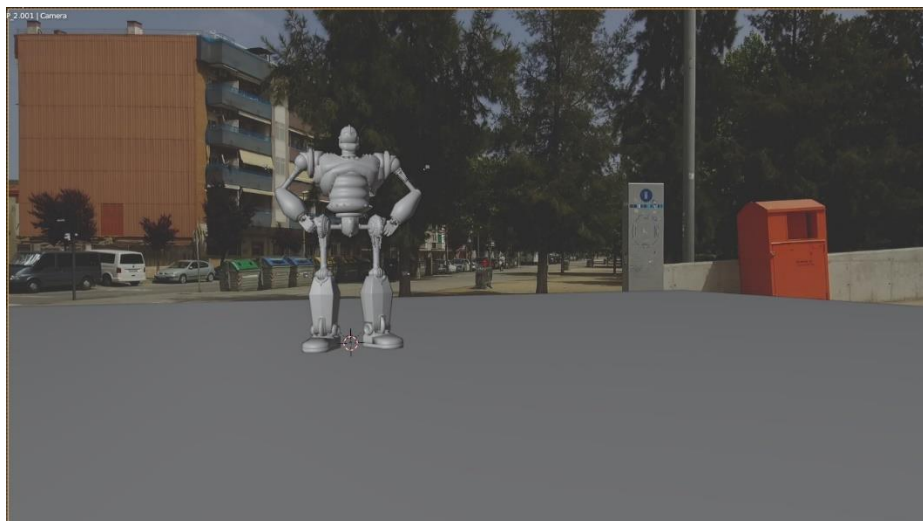


Figura 48. Montaje de set virtual.

3.2.4.5 Evaluación del *matchmoving*

Después de haber completado la cámara y set virtuales es necesario poner a prueba el *matchmoving* para evaluar su correcto funcionamiento, para eso se añade una textura de tablero de ajedrez con una transparencia del 50% para poder ver si hay algún desfase de movimiento entre los cuadrados del tablero y el suelo (ver fig. 49).



Figura 50. Evaluación de *matchmoving* con texturas de tablero de ajedrez.

3.2.4.6 Entrega de escena

Ahora que se ha evaluado apropiadamente el *matchmoving*, y se consiguió el resultado deseado, somos capaces de ir al siguiente paso en el flujo de trabajo que corresponde a la entrega de la escena virtual generada para iluminar y mandar a *render*. Para eso no es necesario, en el caso del desarrollo de este proyecto, migrar a otra aplicación ya que esta tarea puede ser completada perfectamente con Blender.

3.2.5 Iluminación de escena

Para la iluminación de esta escena usaremos dos fuentes de luz, una por medio de una imagen HDR, que fue capturada en la misma locación donde se grabó el vídeo, y la otra por medio de una luz de sol.

3.2.5.1 Creación de imagen HDR

Durante la grabación de vídeo se tomó 3 fotos con diferente exposición: baja, media y alta exposición, con la finalidad de unir las y crear una imagen HDR (ver fig. 50).



Figura 50. Fotos con diferente exposición.

Para poder unir las y generar nuestra imagen HDR se importan las fotografías a Photoshop¹ mediante el menú de *file, scripts, load files into stacks* y aparecerá la ventana de diálogo de *load layers* desde la cual se busca las imágenes a fusionar (ver fig. 51).

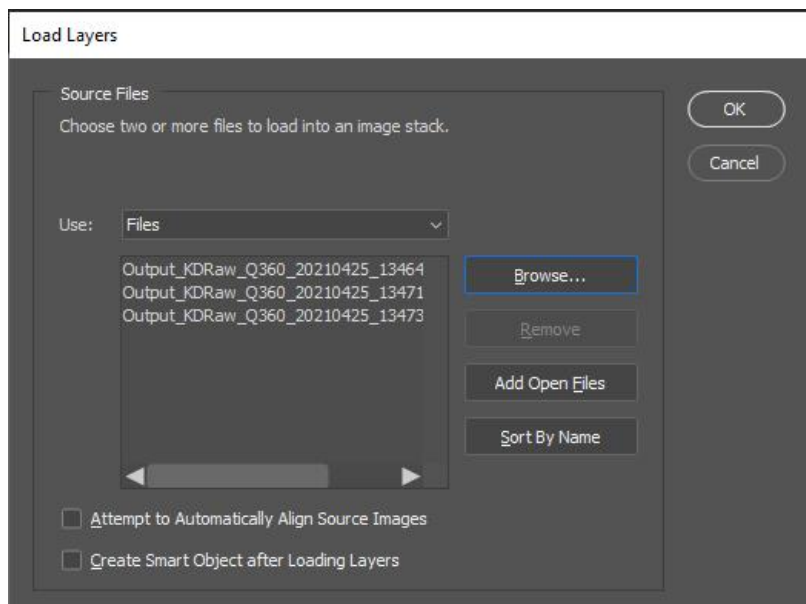


Figura 51. Ventana de load layers

¹ <https://www.adobe.com/es/>

Una vez importadas en Photoshop, se seleccionan todas las capas de las fotos para después en el menú *edit* seleccionar la opción de *auto-blend layers* y, en su ventana de dialogo, se activa la opción de *stack images* (ver fig. 52).

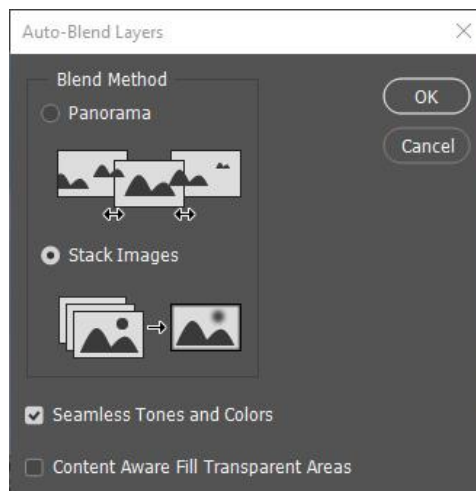


Figura 52. Ventana de Auto-Blend Layers

El resultado de esta operación será una imagen de alto rango dinámico la cual se debe de transforman a imagen de 32 bits, en el menú de *image, mode, 32 bits*, para ser exportada como un archivo de extensión *.HDR¹* (ver fig. 53).



Figura 53. Resultado de fusión de imágenes con diferente rango dinámico.

¹ Extensión de archivo contenedor de imágenes de alto rango dinámico que se utiliza para mejorar el rango de brillo y color de una imagen digital.

3.2.5.2 Importación de imagen HDR a Blender

Para importar la imagen HDR dentro de Blender hay que ir al espacio de trabajo de *shading* y en el *shade editor* elegir la opción de *world*, llegado a este punto se importa la imagen HDR a un nodo de *environment texture*. Para poder rotar la imagen HDR y conseguir que cuadre con la escena hay que agregar dos nodos adicionales, uno de *vector mapping* y un *texture coordinate* (ver fig. 54).

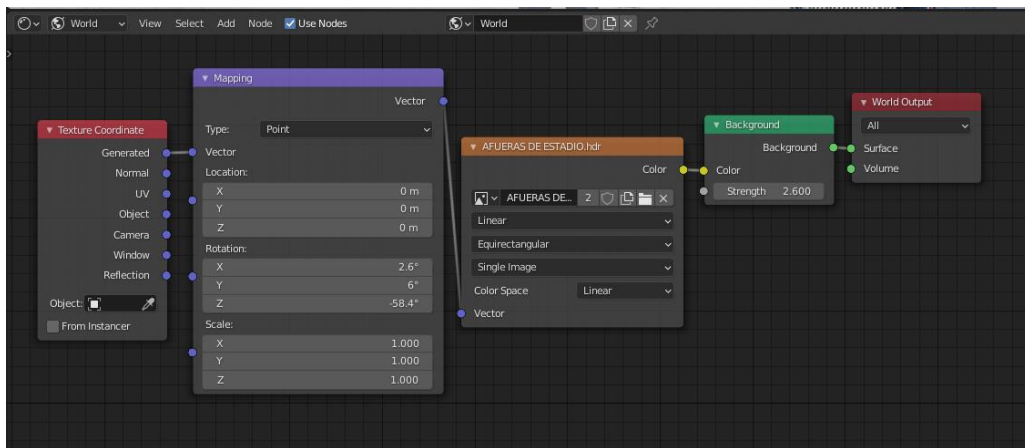


Figura 54. Sistema de nodos para agregar imagen HDR.

3.2.5.3 Iluminación con sun

La iluminación con la imagen HDR sirve para que el modelo adquiera una colorización y reflejos lo más fiel posible al entorno en donde se grabó, pero esta no es suficiente para la correcta generación de sombras, para resolver esto se agrega una luz de sol que, mediante la adecuada configuración, tendrá características muy similares a del sol del día de rodaje y proyectará sombras similares a las vistas en el entorno del vídeo (ver fig. 55).



Figura 55. Sombra proyectada generada por luz sun.

3.2.6 Render de escena

Finalizado el proceso de iluminación de la escena, corresponde configurar los parámetros de *render* y su formato de salida para finalmente poder exportarlo.

3.2.6.1 Configuración del *Render*

Para acceder a los parámetros de configuración del render hay que dirigirse al apartado de propiedades y seleccionar el icono de *render properties* (ver fig. 56).

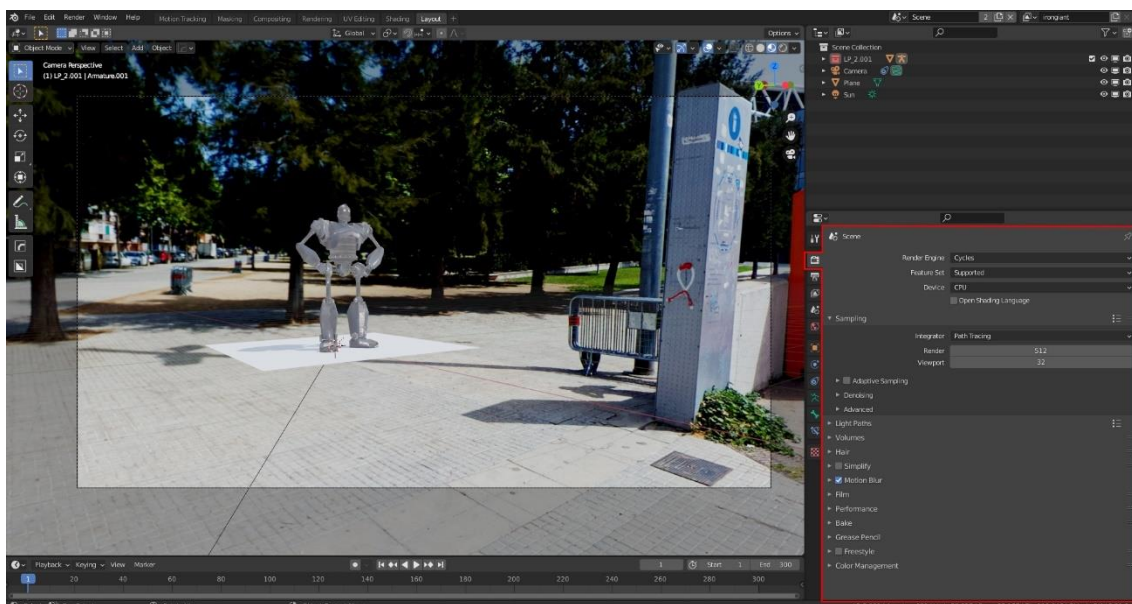


Figura 56 *Render properties*.

Dentro del apartado de propiedades de *render* se procede a configurar el motor de *render* a utilizar, en este caso *Cycles*, motor de render basado en físicas de Blender, luego se configura el *sampling* de render en 512, entre más alto sea el valor del *sampling*¹ mayor será la calidad del render. Para que el render tenga canal Alpha hay que activar, en la sección de *film*, la opción de *transparent*, con esto se asegura que el render poseerá una transparencia que permita mostrar al video de fondo al momento de hacer la integración en postproducción (ver fig. 57).

¹ Proceso por el cual los rayos de luz interactúan con los objetos, *shaders* y materiales de la escena, recolectando información, para luego viajar a la cámara y determinar el color de cada *pixel* en el render.

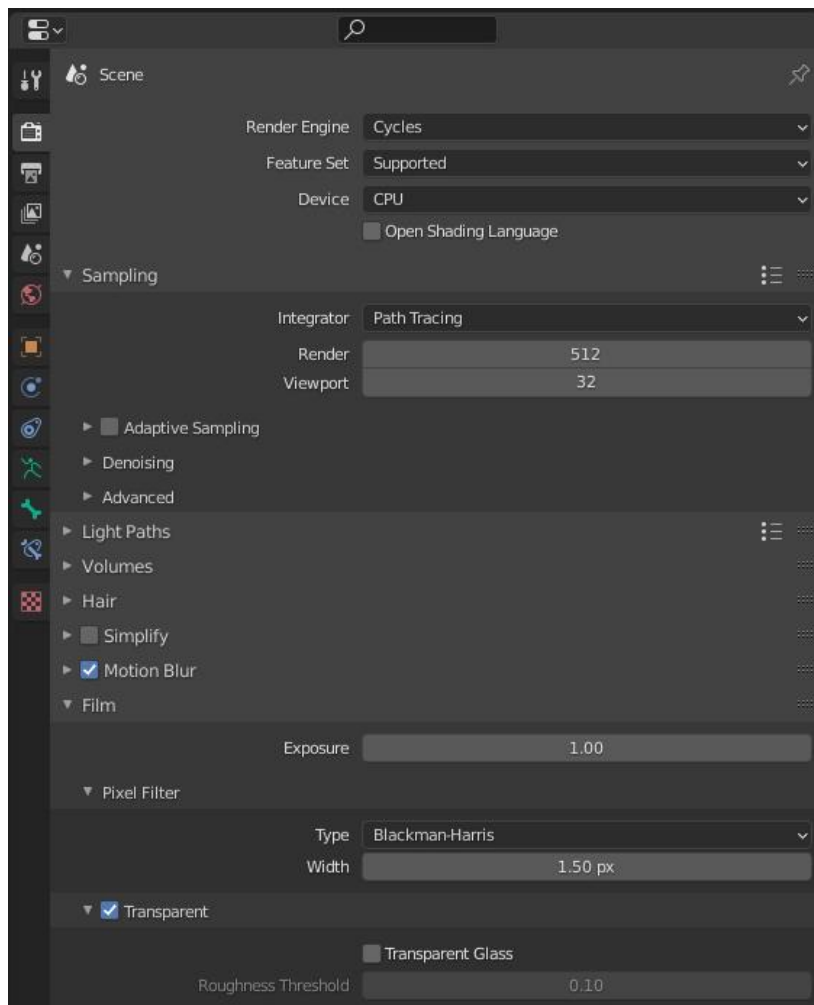


Figura 57. Configuración de render.

3.2.6.2 Configuración del formato de salida

La configuración del formato de salida empieza en el apartado de *output properties* con la selección de la ruta de salida, para después establecer el formato de archivo en OpenEXR MultiLayer¹ con color RGBA² (ver fig. 58).

¹ formato de fichero para imágenes de alto rango dinámico desarrollado como un estándar abierto ampliamente usado en la producción 3D. Es *multilayer* cuando puede alojar capas adicionales de información.

² Siglas de canales rojo, verde, azul y Alpha

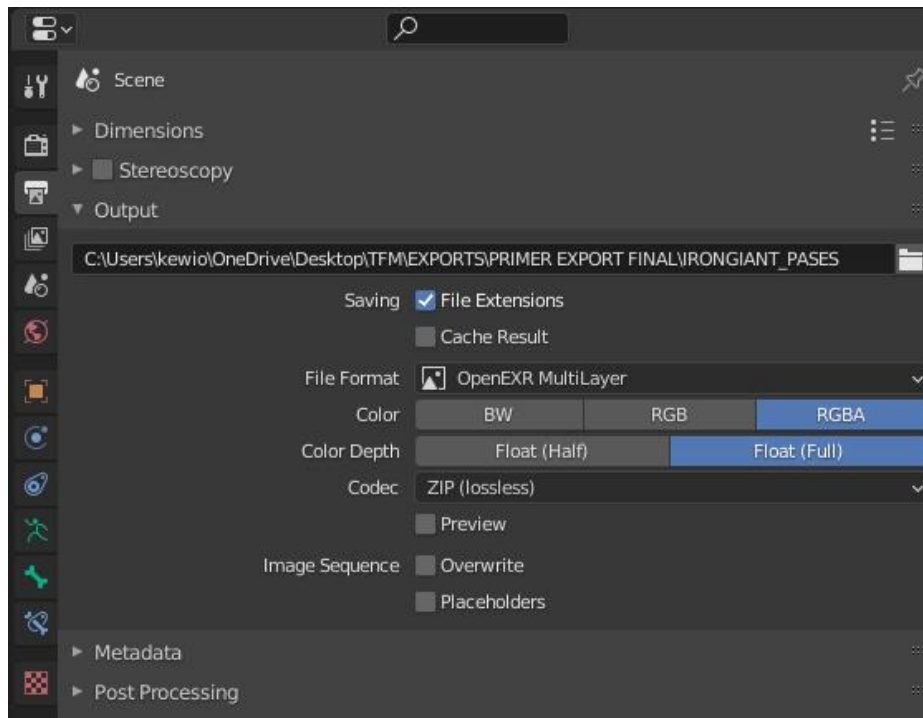


Figura 58. Configuración de render

Luego, en el apartado de *view layer properties*, se configuran las capas que vendrán incluidas en el archivo *OpenEXR MultiLayer* para poder ser más preciso en la postproducción, tales como el *Emission* que registra emisiones de luz de material, el *Ambient Occlusion* y un *Cryptomatte* de objeto que sirve para poder separar por medio de colores partes específicas del render (ver fig. 59).

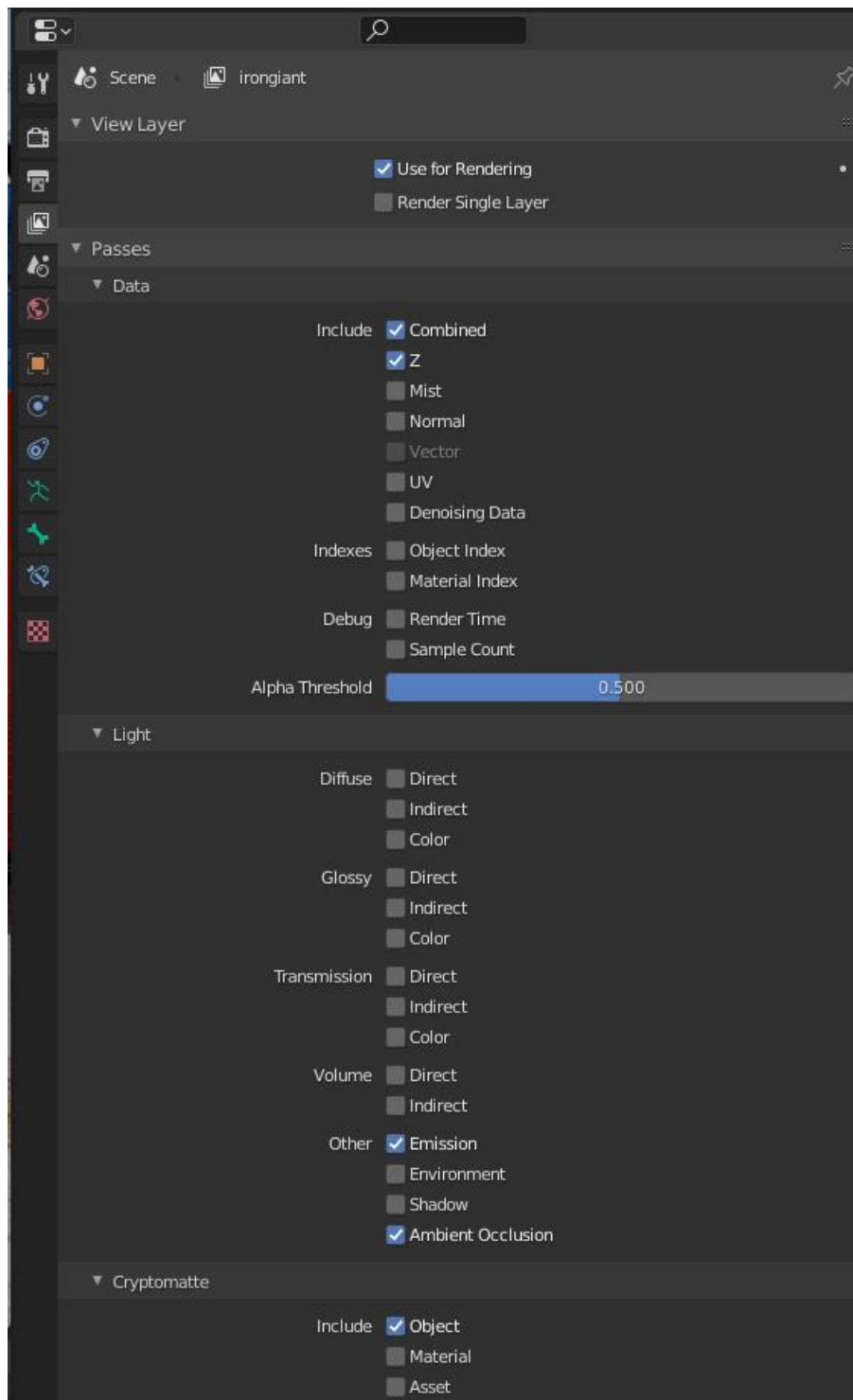


Figura 59 View Layer Properties.

3.2.6.3 Envío de escena a *render*

Ahora que todos los parámetros para el *render* están correctamente configurados solo queda enviar la escena a *render*, para eso se presionan los comandos CTRL + F12 activando así la ventana de *Blender render* que dará inicio al *render* de la escena (ver fig. 60).

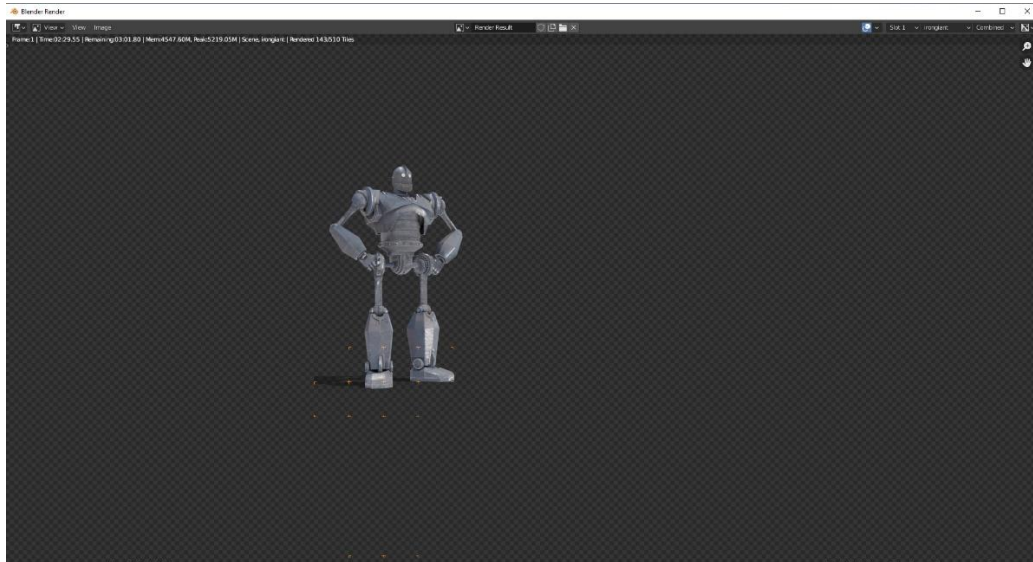


Figura 60 Render de escena.

3.3 Postproducción

En este apartado se tratará la integración de la imagen real de video con el render final de la escena, que se ha generado en la etapa de producción, mediante el *software* de After Effects.

3.3.1 Importación de archivos a After Effects

Para importar los archivos debemos ir al menú de *File, import, File* lo cual abrirá la ventana de *Import File* en donde se debe buscar la ruta del archivo a importar y luego dar *click* en el botón de *import* (ver fig. 61).

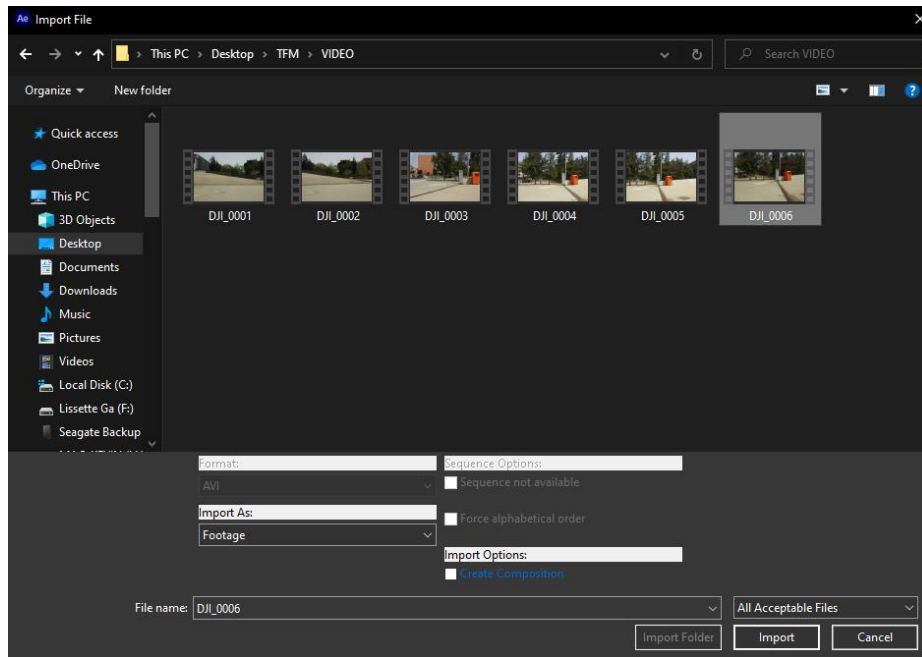


Figura 61. Ventana de Import File

3.3.2 Creación y configuración de composición

Con los archivos finalmente importados es el turno de crear y configurar la composición en donde se va a trabajar, para eso hay que dirigirse a la barra de menú y seleccionar *Composition*, lo cual desplegará un submenú, y elegir la opción de *New Composition*, esto abrirá la ventana de *Composition Settings* en donde se deben de establecer los parámetros de la composición. Para este proyecto se deja el *preset* de HDTV 1080 25 que nos da un formato de video de 1920 x 1080 a 25 frames por segundo y una duración de 12 segundos, que es lo dura el vídeo, por último, se da *click* en *Ok* (ver fig. 62).

Ahora que la composición ha sido creada correctamente se pueden agregar los archivos del video y la secuencia OpenEXR del *render* a la misma.

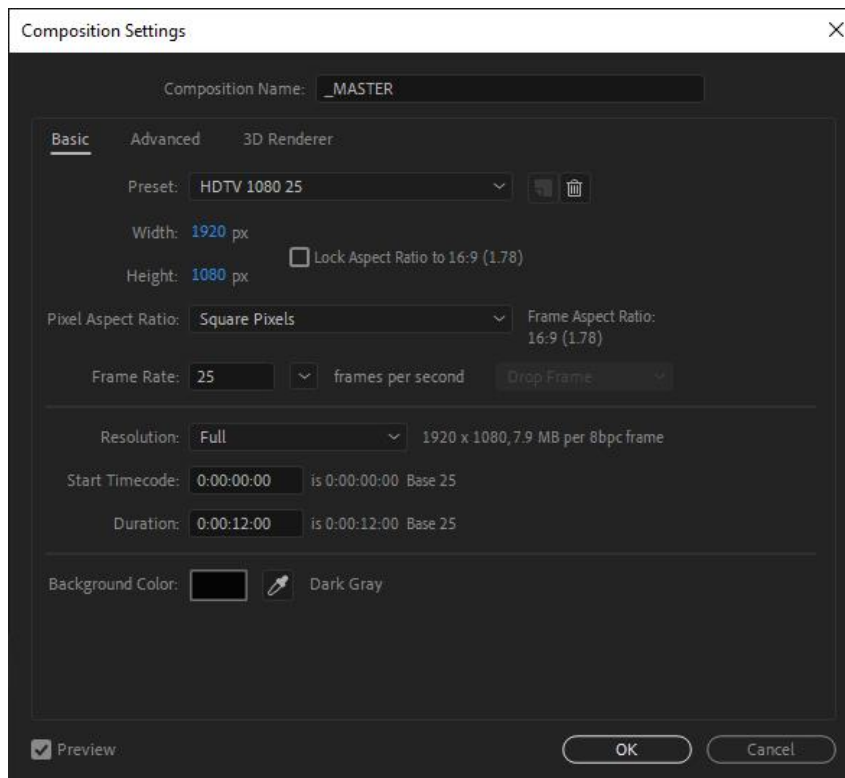


Figura 62. Vista de Composition Settings

3.3.3 Preparación de archivo OpenEXR MultiLayer

Cuando el archivo OpenEXR MultiLayer es importado a la composición mostrará una imagen totalmente negra debido a que tiene muchas capas y hay que elegir cual capa se visualizará, para eso se le agrega a la capa del render el efecto de EXtractoR y se selecciona la capa del render completo (ver fig. 63).

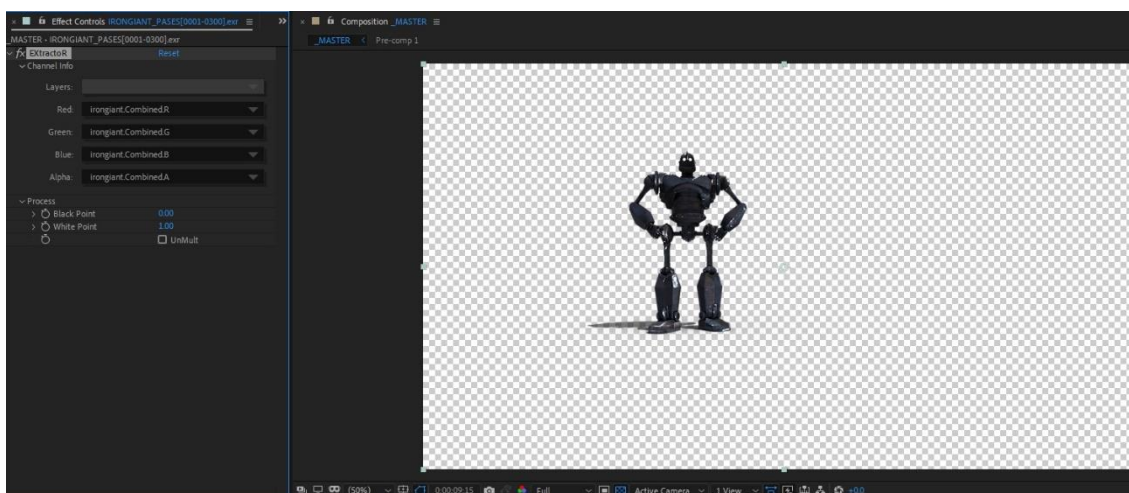


Figura 63. eEfecto de EXtractoR y capa de render

Para visualizar otra capa del archivo OpenEXR MultiLayer simplemente se la selecciona en el efecto de EXtractoR (ver fig. 64).

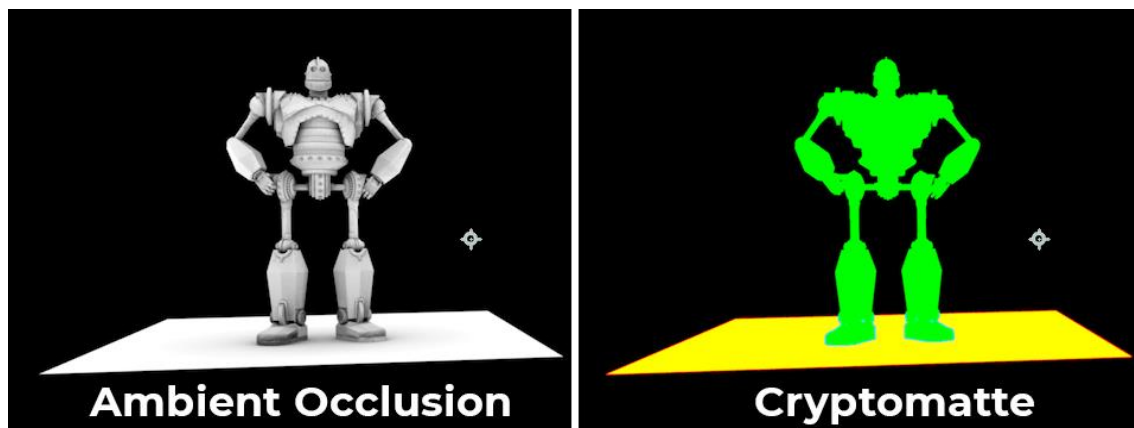


Figura 64. Capas de Ambient Occlusion y Cryptomatte de archivo OpenEXR MultiLayer

3.3.4 Colorización de Iron Giant

Por medio de la colorización se consigue que el render adquiera la misma tonalidad y exposición presente en el vídeo, para lograrlo hay que agregar a la capa del *render* un efecto de *color profile converter* y activar la casilla de *Linearize Input profile* para que la capa muestre el perfil de color con el que se está trabajando el proyecto, ya que por defecto nos muestra el perfil de color del *render*, luego se agrega el efecto de *Lumetri color* para empezar a colorizar la capa. Dentro del efecto de *Lumetri color* se configura, en el apartado de *Basic correction*, los tonos de las luces, sombras, exposición y contraste en función del vídeo y por último se envía la temperatura de color del render hacia los colores cálidos ya que el vídeo presenta esa tonalidad, luego de que la capa ha sido colorizada se procede a aislar al modelo para quitar la sombra y colorizarla por separado, esto se hace agregando otra capa del archivo OpenEXR MultiLayer con el efecto de EXtractoR seleccionando la capa de *Cryptomatte* y agregando el efecto de *Color Range* para seleccionar el color amarillo del suelo, por último a la capa del render se le hace un *Alpha matte* de la capa del *Cryptomatte*. Todo este proceso se repite para colorizar y aislar la sombra y permitir una edición más controlada de los elementos (ver fig. 65).

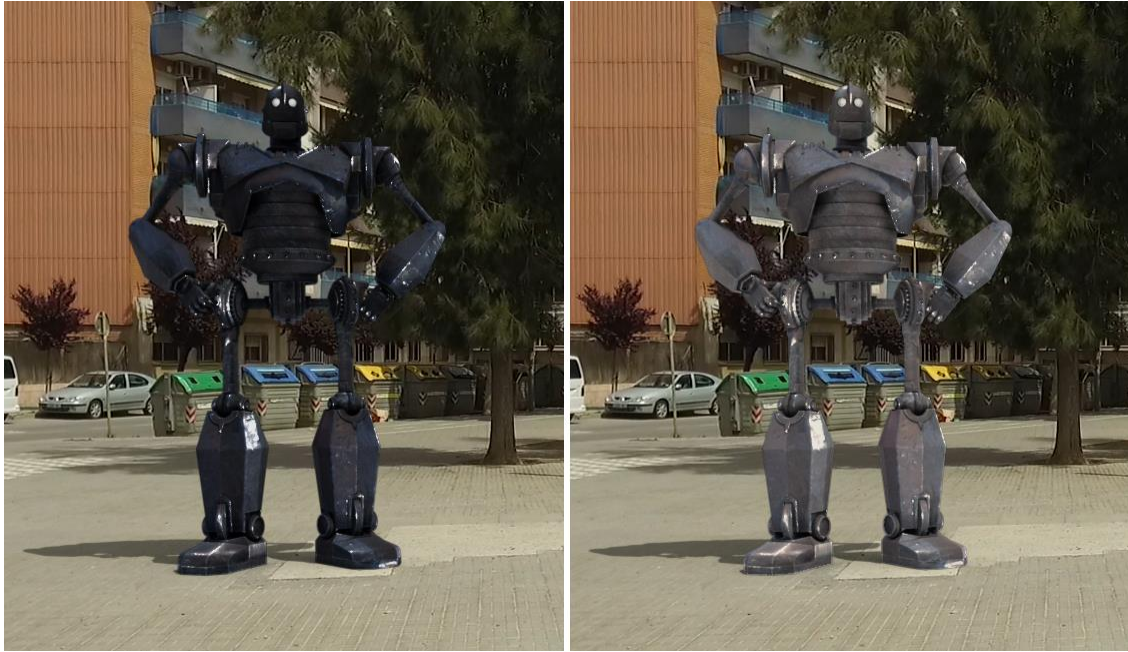


Figura 65. A la izquierda capa sin colorizar, a la derecha capa colorizada.

3.3.5 Exportación de composición

Con la integración del render y el video completa solo queda mandar a exportar el vídeo, para eso hay que ir al menú de composición y dar *click* en *Add to Render Queue*, en donde solo queda configurar los parámetros de exportación deseados y elegir la ruta de destino, finalmente se da *click* en *Render*.

4. Conclusiones

A nivel personal la realización de este proyecto ha sido un reto que he disfrutado, me ha hecho pensar e investigar tantos campos desconocidos o vagamente entendidos por mí y sobre todo me ha servido para reafirmar mi vocación hacia el mundo de la producción 3D, efectos especiales y postproducción.

El objetivo que más dificultades presento fue el de *matchmoving* ya que trasladar la teoría a la práctica en este caso en particular requiere de mucho entrenamiento, incluso en las fuentes consultadas se me advertía de esto, debido a que para convertirse en un buen *matchmover* hay que crear una especie de intuición que solo se logra desarrollarse a base de ensayo y error. Como resultado de la práctica de *matchmoving* también me llevo la experiencia para futuros proyectos en los que consideraría colocar puntos de rastreo en la escena y luego limpiarlos en postproducción.

Una reflexión muy importante, en cuanto al flujo de trabajo, es que hay que crear buenas bases teóricas, recurriendo a textos, informes, memorias o documentación web, y no solo concentrarse en el desarrollo técnico de las herramientas ya que cuando un profesional se especializa en la herramienta sin entender del todo las bases y el porqué de cada acción, luego le resulta muy difícil migrar a otro software y considero que esto es un grave error. Un buen postproductor deber desarrollar la adaptabilidad de softwares, flujos de trabajo y estar en constante investigación de lo que ocurre en el mercado.

El presente proyecto me ha servido para poner a prueba el potencial de la herramienta de Blender, considero que es un software que pronto ocupara un lugar importante dentro del flujo de trabajo de pequeños y medianos estudios ya que ofrece un amplio espectro de trabajo en la producción de proyectos audiovisuales. Junto con Blender, existen algunos softwares de uso gratuito que están convirtiéndose en estándares, lo cual considero es una oportunidad muy grande para los profesionales que estamos empezando en este mundo de los efectos especiales y producción 3D y no hay que desaprovecharla.

5. Referencias

Vaughan, W. (2011). [digital] Modeling. Consultado el 26 de mayo de 2021, en <https://learning.oreilly.com/library/view/digital-modeling/9780321712622/copy.html>

Bloch, C. (2013). The HDRI Handbook 2.0. Consultado el 10 de Junio de 2021, en <https://learning.oreilly.com/library/view/the-hdri-handbook/9781457179303/ch03.html#ch03>

Cooper, T. (2015). HDR Photography: From Snapshots to Great Shots. Consultado el 10 de Junio de 2021, en <https://learning.oreilly.com/library/view/hdr-photography-from/9780134181240/ch01.html#ch01lev1sec5>

Kumar, A. (2020). Beginning PBR Texturing: Learn Physically Based Rendering with Allegorithmic's Substance Painter. Consultado el 26 de Junio de 2021, en https://learning.oreilly.com/library/view/beginning-pbr-texturing/9781484258996/A492314_1_En_12_Chapter.html

Dobbert, T. (2012). Matchmoving: The Invisible Art of Camera Tracking, 2nd Edition. Consultado el 3 de Julio de 2021, en <https://learning.oreilly.com/library/view/matchmoving-the-invisible/9781118529669/c06.html>

Udemy. (2021). Substance Painter para principiantes nivel 0 [Video]. Consultado en <https://www.udemy.com/course/substance-painter-postapocaliptico/learn/lecture/21582518#questions>

Blender 2.91 Manual de Referencia — Blender Manual. (2020). Consultado en 6 de junio de 2021, en <https://docs.blender.org/manual/es/2.91/>

Bird, B. (1999). The Iron Giant [Film]. United States: Warner Bros. Feature Animation.