

TRABAJO FIN DE MÁSTER

Título:

**DISEÑO Y CREACIÓN DE UN ENTORNO 3D
PARA UN VIDEOJUEGO**



Autora:

Inés Miguel Alonso

Tutor:

Mario Alaguero Rodríguez

**MÁSTER UNIVERSITARIO EN COMUNICACIÓN Y
DESARROLLO MULTIMEDIA**

Universidad de Burgos
Facultad de Humanidades y Comunicación

2021



ANEXO III

INFORME DEL DIRECTOR/A PARA LA PRESENTACIÓN DEL TRABAJO DE FIN DE MÁSTER

D./D.ª Mario Alaguero Rodríguez

Director/a del Trabajo Fin de Máster titulado “Diseño y creación de parte de un escenario 3D para un videojuego”, realizado por el alumno/a D./D.ª Inés Miguel Alonso,

INFORMA:

POSITIVAMENTE

NEGATIVAMENTE (motivar)

.....
.....
.....

la defensa del Trabajo Fin de Máster correspondiente al Máster Universitario en Comunicación y Desarrollo Multimedia.

Burgos, a 6 de juliode 2022

EL/LA DIRECTOR/A

Fdo.: Mario Alaguero Rodríguez

Agradecimientos

Después de estos meses de intenso trabajo, dedico este apartado de la memoria de mi Trabajo de Fin de Máster a todas las personas que me han acompañado y apoyado en esta etapa de realización y aprendizaje tanto a nivel académico como a nivel personal.

En primer lugar, me gustaría agradecer a mis compañeros de trabajo por su enorme paciencia, colaboración y apoyo siempre que lo necesitaba.

En especial, quería expresar la gratitud a mi tutor por su apoyo, consejos y lecciones, sin las que habrían sido posibles el desarrollo de este proyecto.

Por último, agradezco a mis padres y hermana por haberme acompañado siempre, incluyendo los momentos difíciles, y haberme ayudado cuando lo he necesitado.

ÍNDICE

1. Introducción	5
2. Objetivos	6
3. Marco teórico	7
3.1. El hierro y el acero de la Primera y Segunda Revolución Industrial	7
3.2. Videojuegos <i>indie</i>	8
3.3. Videojuegos de carreras	9
3.4. Creación de videojuegos	14
3.4.1. Modelado de videojuegos	15
3.4.2. Texturizado de videojuegos	18
3.5. Software de modelado y texturizado 3D.....	20
4. Metodología	26
4.1. Preproducción	26
4.1.1. Documentación	26
4.1.2. Selección del software	27
4.1.3. Planificación	28
4.1.4. Trabajo en equipo	29
4.2. Producción	29
4.2.1. Diseño	29
4.2.2. Modelado	30
4.2.3. Texturizado	30
4.2.4. Renderizado	32
5. Desarrollo.....	33
5.1. Preproducción	33

5.1.1. Referencias para el diseño y modelado del escenario.....	33
5.1.2. Referencias para el texturizado del escenario.....	36
5.2. Producción.....	37
5.2.1. Diseño del escenario.....	37
5.2.2. Modelado.....	38
5.2.3. Texturizado.....	49
6. Conclusiones.....	55
6.1. Análisis del esfuerzo realizado.....	56
7. Resumen.....	58
8. Abstract.....	59
9. Bibliografía y fuentes.....	60
10. Índice de imágenes.....	69
11. Anexos.....	71
Anexo 1. Esquema de las etapas y planificación.....	71
Anexo 2. Flujograma.....	72
Anexo 3. Bocetos del escenario.....	73
Anexo 4. Material multimedia.....	77

1. Introducción

Este Trabajo Fin de Máster se centra en el modelado de una parte del escenario del videojuego de carreras *Fastastic Roads*. Dicho trabajo forma parte de un proyecto más amplio en el que participan el centro de investigación ÍTACA (Centro de Innovación y Tecnología en Videojuegos y Comunicación Audiovisual), cuyo objetivo en este proyecto es la creación del videojuego completo.

El videojuego *Fastastic Roads* se ambienta a finales del siglo XIX y principios del siglo XX, centrándose en la Revolución Industrial. El videojuego pretende dar a conocer a las cuatro protagonistas, pilotos de los vehículos, por su labor e influencia en el mundo del automóvil e ingeniería durante los siglos XIX y XX. Estas cuatro mujeres son Valentina Tereshkova, Amelia Earhart, Pilar Careaga y Basabe, y Bertha Benz. Sus respectivos vehículos están basados en la cosmonave Vostok 6, el avión Lockheed Vega 5b, un ferrocarril de época y el coche Benz Patent-Motorwagen Typ III. Tanto los personajes como sus vehículos ya fueron diseñados y creados por la alumna Claudia Calvo Cuetos para su Trabajo de Fin de Máster en 2020. Su trabajo se utilizará como base para establecer los fundamentos de este presente proyecto.

El objetivo principal de la realización de este Trabajo Fin de Máster consiste en diseñar, modelar y texturizar parte de un escenario para que se puedan llevar a cabo las carreras. Por tanto, el proyecto requiere una fase de diseño tanto de la pista como de los elementos que componen el escenario y la producción de todo lo planificado de manera funcional para que la programación del videojuego se ajuste a las fases previstas y los resultados sean óptimos encontrando un equilibrio entre calidad de imagen y rendimiento del videojuego.

Con el fin de llevar a cabo este proyecto, se requiere una detallada planificación para optimizar el tiempo de ejecución del mismo y también se necesita una buena coordinación del equipo de trabajo, pues existen muchas partes implicadas que tienen que estar en constante sincronía.

Además, es necesario buscar una buena documentación para que el diseño y los elementos del escenario sean acordes a la época a la que se basa el videojuego. También estas referencias son imprescindibles en la narrativa del proyecto. El formato de referencia será el Sistema Harvard-APA.

2. Objetivos

El objetivo principal en este Trabajo Fin de Máster es el diseño de parte de los elementos y escenario para el videojuego *Fastastic Roads* y la creación de los mismos en tres dimensiones mediante ordenador con el objetivo de que funcionen correctamente en el videojuego. También, se pondrán de manifiesto los conocimientos adquiridos en el Máster en Comunicación y Desarrollo Multimedia de la Universidad de Burgos.

La parte del escenario que se va a modelar es una fábrica de hierro y acero. A esta fábrica la componen tres edificios: uno donde se trocea el mineral extraído de la montaña que se encuentra en el escenario y los otros dos donde se procesa el material anterior.

A este objetivo lo complementan otros secundarios. En cuanto al diseño, se deben fijar unas pautas a seguir por todos los diseñadores del equipo en la fase de modelado y en la de texturizado, con la finalidad de establecer un estilo uniforme en todo el ambiente del videojuego. Por ello, las referencias sobre las que se inspiren han de ser claras y precisas para establecer un estilo único y reconocible para el videojuego.

Añadido a lo anterior, tanto el diseño del circuito como el modelado de los elementos y texturizado posterior habrá de estar optimizado en cuanto a narrativa como a nivel técnico. Esto implica que no solo los elementos que componen el escenario han de tener un sentido en relación con los personajes y su historia, sino que además deben estar modelados según lo establecido debido a la estética visual y a para que el rendimiento del videojuego sea el óptimo.

Por último, a nivel personal el objetivo es mejorar y adquirir nuevas habilidades y conocimientos en el ámbito del diseño, modelado 3D, texturizado y diseño de videojuegos con la finalidad de aplicarlos en un futuro profesional.

3. Marco teórico

En este apartado se exponen las investigaciones previas y referencias utilizadas para la planificación y desarrollo del proyecto. Los puntos que se tratan son la historia y empleo del hierro y el acero durante la Segunda Revolución Industrial, los videojuegos *indie*, los videojuegos de carreras, el modelado de videojuegos, su texturizado y el *software* seleccionado para su desarrollo. Esta investigación tiene una gran importancia a la hora de comenzar a planificar y diseñar, para que luego el trabajo se lleve a cabo de una manera satisfactoria y cómoda para todos los miembros del equipo.

3.1. El hierro y el acero de la Primera y Segunda Revolución Industrial

El término “Revolución Industrial” se utilizó por primera vez a principios del siglo XIX para resaltar la evolución de la mecanización industrial en Francia, en comparación con la revolución de 1789 (Cameron, 1985). En el siglo XX para lograr el crecimiento del país el objetivo fue la industrialización. Sin embargo, la primera revolución industrial surgió en Inglaterra en el siglo XVIII, convirtiéndose en la representación del avance de la economía moderna (Berzosa, 2008). Durante los siglos XIX y XX tuvo lugar la Segunda Revolución Industrial, siendo característico el uso del hierro en su mayor parte en el siglo XIX por ser más resistente al fuego y el hormigón armado en el siglo XX (Bernabeu, 2007).

No obstante, antes del descubrimiento del hormigón para realizar construcciones, tuvo lugar la evolución del acero como sustituto del hierro. Durante el siglo XVIII el acero era más caro que el hierro y se utilizaba para fabricar piezas concretas de armas. Las máquinas, puentes y otras estructuras estaban construidas con hierro forjado o fundiciones (Martínez, 1997).

A principios del siglo XIX, surgió maquinaria más compleja en las fábricas, al igual que métodos para obtener acero. Uno de ellos era el de Chenot. Este consistía en seleccionar el mineral, trocearlo y calentarlo para conseguir hierro puro. El resultado era una esponja a la que se clasificaba según la cantidad de carbón que tuviera. A través de un ventilador se separaba definitivamente y luego la esponja se forjaba o se transformaba en acero (Uriarte, 1998).

En la segunda mitad del siglo XX, se consiguió transformar impurezas del hierro tras su fundición (arrabio) en acero en el convertidor Bessemer (Paíno, 2004). Estas impurezas se oxidan por la reacción entre el óxido de hierro y el silicio y manganeso que contiene este mineral. La oxidación del silicio eleva la temperatura y llega a formarse dióxido de carbono.

Finalmente, el hierro se ha oxidado. Luego, se pasaba a moldes para crear chapas, tubos y perfiles laminados. A través de otras técnicas y aplicaciones, sustituyeron este método por otros más eficientes. Así comenzó la producción en masa de acero e Inglaterra se convirtió en la primera potencia industrial en el mundo (Grubler, 1995).

En las fábricas la principal fuente de energía era el carbón. Este era el combustible de las máquinas a vapor tanto para maquinaria de fabricación como de transporte. El acero se comenzó a utilizar en estos campos y Estados Unidos se logró posicionar como principal productor de acero y automóviles (Grubler, 1995). Además, no solo la Revolución Industrial trajo consigo cambios económicos e industriales, sino también sociales. Los propietarios de las fábricas tenían que mantener las máquinas siempre activas, dado que era muy costoso que estas estuvieran paradas. Por ello, se introdujeron el montaje en cadena y el formato “cronometrado” en el que se establecían horarios a los trabajadores, iniciando el trabajo regular y las desigualdades entre clases sociales (Stearns, 2018).

3.2. Videojuegos *indie*

Aunque no existe una definición exacta, un videojuego *indie*, o también conocido como videojuego independiente, se puede definir como un videojuego creado por una pequeña entidad o individuo(s) autónomos. Suelen ser videojuegos de presupuesto inferior, no ponen límites a la creatividad y la falta de presupuesto les lleva a innovar en el diseño (M. Simón Tudanca en KAZ: Pushing the Virtual Divide, 2014). En palabras de Chris Dahlen, cofundador de Kill Screen (Indie game: The Movie, 2011): “son cualquier juego en el que un pequeño equipo o individuo trabajó fiel a su propia visión, algo que deseaban hacer, programar y terminar”.

Por el contrario, los videojuegos Triple A (también conocidos como videojuegos AAA) son superproducciones de las grandes empresas del sector y, por ello, cuentan con un elevado presupuesto. “Los juegos grandes tienen otras metas, entretener. Sin embargo, los juegos independientes a menudo son una persona que dice ‘quiero un juego que sea como esto, que trate sobre esta cosa especial’” (Gus Mastrapa en Indie game: The Movie, 2011). Además, los videojuegos *indie* no suelen tener apoyo financiero de editores.

Los primeros videojuegos *indie* se desarrollaron para ordenadores. Esto se debió a que, en la década de los 90, tenían la concepción de que los ordenadores eran más familiares y más accesibles que el resto de plataformas de videojuegos. También, el *software shareware*, un programa que se distribuye gratis para que se pruebe, resultó ser de gran ayuda para que se pusieran a prueba los proyectos *indie* (PCGAMER, 2017). La expansión de Internet a comienzos del siglo XXI fue otro de los factores de la expansión de los videojuegos independientes. Se popularizaron los creados en Adobe Flash, que consistían en videojuegos cortos, entretenidos y simples. En 2003 se lanzó la plataforma Steam como distribuidora digital de videojuegos. Por ello, unido a la expansión de Internet, los videojuegos *indie* fueron más accesibles para un mayor número de jugadores (Culture of gaming, 2017).

3.3. Videojuegos de carreras

Un videojuego de carreras plantea una competición en un circuito donde tiene el protagonismo la habilidad de conducir. Los jugadores toman el control de sus vehículos, los cuales no solo pueden tratarse de coches o motos, sino que pueden ser vehículos aéreos, navales, acuáticos o incluso otros más creativos. El jugador tiene como objetivo principal alcanzar la meta tras recorrer el circuito, en las vueltas pertinentes o en el menor tiempo posible, en la primera posición, habiendo superado al resto de los jugadores.

Otra categoría similar a la de los videojuegos de carreras es la de simulación. Esta tiene como objetivo la representación lo más cercana a la realidad del control del vehículo. Por el contrario, los videojuegos de carreras prioriza la experiencia del jugador, es decir, que sea entretenido (Apperley, 2006).

El primer videojuego considerado del género de carreras es *Atari's Space Race* (1973) para máquinas de arcade. Dos jugadores competían con naves espaciales por la primera posición mientras evitan chocar con obstáculos. *Gran Trak 10* fue lanzado por Atari un año después (Kent, 2016). Este videojuego de arcade consistía en que los jugadores condujesen un coche por un laberinto a contrarreloj mientras evitaban chocar con los bordes. Estos videojuegos tuvieron sus precursores en máquinas de salones recreativos. A finales de los años 50 destacaba *Mini Drive*, desarrollado por Kasco, que era un juego electro-mecánico de carreras de coches. La pista era una cinta transportadora física y se trataba de competir con otros coches sin chocar contra ellos (Shmuplations, 2001).



Figura 1. *Mini Drive* (1959). Fuente: Shumplations.

En la década de los 70, la misma empresa que desarrolló *Mini Drive*, Kasco, lanzó *Indy 500* para la consola Atari 2600. Este ofrecía tres modos de juego: carrera contrarreloj o hasta completar 25 vueltas, dos jugadores compiten acumulando puntos y, por último, una competición de roles entre dos jugadores (MeriStation, 2012).

Atari desarrolló el videojuego *Night Driver* en 1976. En pantalla se mostraban dos líneas paralelas que señalaban la carretera. No había coche dentro del propio videojuego, sino que estaba impreso y se colocaba bajo la pantalla de la máquina de arcade. Se jugaba desde la perspectiva de primera persona en un ambiente nocturno. A finales de la década de los años 70, creció en popularidad la empresa SEGA. En 1979, lanzaron *Monaco Grand Prix*, el cual ofrecía coches de la competencia yendo de lado a lado obstaculizando el paso del jugador y los vehículos alcanzaban una mayor velocidad. El segundo videojuego de carreras de SEGA fue *Turbo* (1981), más avanzado que su precursor *Night Driver*: ofrecía un mayor realismo y una optimización del rendimiento gráfico, lo que provocaba un mayor movimiento y profundidad (Herraez, 2011).

Namco, la empresa creadora del videojuego *Pac Man*, desarrolló *Pole Position*, gracias a su referente *Turbo*, generando una experiencia real de las carreras de Fórmula 1. El videojuego situaba al jugador detrás del coche y su nivel gráfico era excelente por sus colores llenos de luminosidad, nitidez de las imágenes y el detalle. (Electronic Fun, 2014). Además, el

videojuego trasladaba al jugador al autódromo de Fórmula 1 *Fuji Speedway* en Japón e incluía rondas de clasificación.



Figura 2. *Turbo* (1981). Fuente: Sega Retro.



Figura 3. *Pole Position* (1982). Fuente: Time Graphics.

En los años 80, el físico Geoff Crammond desarrolló videojuegos e introdujo las coordenadas mapeadas en tres dimensiones para incorporarlos a objetos en movimiento con exactitud, aunque la calidad gráfica no mejoró. Este aspecto se optimizó en el videojuego *Test Drive* (1987), que incluso ofrecía un sonido más limpio, a pesar de que puede considerarse también como un videojuego de simulación de coches (Candil, 2011). Además, Crammond desarrolló el videojuego *Formula One Grand Prix* (1991). Este era realista para la época, debido a que incluía diversos modos de juego como diversas maneras de competición, práctica de los

circuitos, simulación de los campeonatos de Fórmula 1. El teclado se podía calibrar como preferiera el jugador e incluso permitía incorporar ciertas ayudas que mejoraban características del coche. Además, el videojuego presentaba mejoras en las físicas y en la calidad gráfica.

No obstante, este realismo al que aspiraban todas las empresas desarrolladoras era totalmente exitoso. Por ello, en los finales de los años 80 se volvió a priorizar la experiencia del jugador. *Buggy Boy* (1985) es un ejemplo de videojuego que combinaba las carreras de coches con evitar obstáculos y saltar (Videojuegos Horacio, s.f.).

El videojuego de carreras japonés *Out Run* se lanzó en 1986. Este es considerado como uno de los primeros videojuegos de su género que ofrecía un mundo abierto más allá del propio radio del circuito de carreras. Además, contaba con unos gráficos avanzados para aquella época, con lo que lograba crear una nueva sensación de velocidad a los jugadores. Esto se conseguía transmitir gracias a la inspiración de uno de sus creadores, Yu Suzuki, el cual experimentó las sensaciones de conducir por Europa (Concept Car Credit, 2019). Inicialmente, se lanzó para máquinas de arcade y fue el primero en dejar escoger al usuario la música que sonaría de fondo. Atari, por su parte, innovó con su videojuego *Hi-Way* (1975) al crear una autopista de desplazamiento (la carretera se desliza de lado a lado de la pantalla), lo que influyó a la industria de los videojuegos de carreras en cuanto a esta nueva tecnología.

Namco lanzó en 1988 el primer videojuego de carreras que utilizaba gráficos en tres dimensiones. Este era *Winning Run* que se jugaba en primera persona y simulaba carreras de Fórmula 1. Asimismo, proponía dos niveles de dificultad que afectaban a la velocidad y a los obstáculos y la máquina de arcade se movía según la dirección del jugador (Ward, 2019).



Figura 4. *Out Run* (1986). Fuente: Mod DB.



Figura 5. *Winning Run* (1988). Fuente: Xtreme Retro.

En 1992 Super Nintendo lanzó *Super Mario Kart*. Este fue un juego que alcanzó el éxito debido a la creatividad de los elementos del diseño de los escenarios y las habilidades que se podían adquirir durante las carreras. Una de las novedades que se empezaron a incluir a partir de entonces fue la acumulación de daños en el coche. Al colisionar, su eficiencia iba disminuyendo y obligaba al conductor a parar en boxes. El videojuego *Nascar Racing*, desarrollado por Papyrus en 1994, fue el primero en incluir la acumulación de daños en el coche. La inteligencia artificial (IA) también se incluyó en *Nascar Racing*. Treinta y dos eran los coches que estaban controlados por el ordenador y cada uno tenía unas características específicas (Palmieri, 2019). Hasta la actualidad, esta inteligencia artificial se ha ido mejorando hasta el punto de que esta se adapta al jugador en términos de dificultad. También en el año 1994, nació *Need for speed*, uno de los videojuegos de carreras de mundo abierto más exitosos. Craig Sullivan fue su creador, el cual lo define (en Concept Car Credit, s.f.) como “un videojuego que solo se trata de conducir el coche”, siguiendo a su predecesor *Out Run*, el cual pretendía que el jugador condujera por placer.

El competidor de este videojuego, *Gran Turismo*, fue lanzado en Play Station, no introdujo novedades. Simplemente, tomó de partida el mundo abierto, casi siendo una simulación por su gran realismo y calidad de los detalles. Intentó aproximarse a la realidad no solo de manera gráfica, sino también con sonidos, música e incorporando rasgos de conducción únicos en cada automóvil. Sin embargo, para esta superproducción, desarrollaron primero *Motor Toon Grand Prix*, un videojuego similar a *Super Mario Kart*. Sobre él realizarían pruebas para posteriormente lanzar *Gran Turismo*. Este influyó en términos de jugabilidad y nivel gráfico. Su éxito fue debido al realismo y a las características de cada automóvil, que le dan personalidad. La esencia del ambiente y las sensaciones es lo imprescindible para el creador del videojuego, Kazunori Yamauchi, superando la importancia de las medidas y grabaciones

de un circuito cuando este se tiene que analizar para su posterior desarrollo en el videojuego (Kazunori Yamauchi en KAZ: Pushing the Virtual Divide, 2014).



Figura 6. *Motor Toon Grand Prix* (1994). Fuente: Zonared.



Figura 7. *Gran Turismo* (1998). Fuente: Gran-turismo.

3.4. Creación de videojuegos

La creación de videojuegos se compone de una serie de fases. La primera de ellas es la preproducción, donde se plantea el concepto de juego: el género, la historia, se crean los primeros bocetos y se define cómo se va a jugar. Luego, en la etapa de producción se diseña el juego a nivel artístico, mecánico (interacción de los personajes en el entorno) y técnico (la programación). Además, se define el motor de juego donde se decide cómo se representan los elementos del juego y cómo se va a interactuar con ellos (Martín & Ballejo, 2012). La última etapa es la de postproducción que consiste en la puesta del videojuego en el mercado (Manrubia, 2014).

3.4.1. Modelado de videojuegos

El modelado es una de las fases de producción de un videojuego.

El modelado en tres dimensiones (3D) es la creación de un objeto tridimensional a través de un *software*. Este proceso se puede realizar a través de varias técnicas. Una de ellas es el modelado de caja que parte de un objeto geométrico al que se le subdivide para crear formas más complejas. Como complemento se utiliza el método de modelado de bordes que genera primero el contorno de las figuras. Otra de las técnicas es utilizar las curvas *Nurbs* (son curvas que se generan a partir de sus puntos de control) que se emplean para crear superficies. Por otro lado, para crear paisajes rápidamente, existe el modelado procedural que funciona automáticamente gracias a algoritmos y a los parámetros definidos por el usuario. Otros tipos de modelado son la escultura digital (realiza una simulación de la acción de modelar a mano), el modelado basado en imágenes (parte de una representación en dos dimensiones de un objeto desde varios ángulos para crearlo en 3D a través de algoritmos) y el escaneo en tres dimensiones que funciona de manera automática, gracias a los datos extraídos desde los distintos ángulos de un objeto o persona a través de escáneres especializados (3D ACE, 2017).

Como caso concreto, para el modelado del videojuego de carreras *Gran Turismo*, se utilizó el origami moderno como proceso previo a la creación de los coches en tres dimensiones. Los artistas emplearon el patrón de las dobleces para generar la esencia abstracta de los vehículos. Esto es porque existe una conexión entre los algoritmos de las mallas tridimensionales y los del patrón de las dobleces de origami, debido a la similitud de posiciones y ángulos de las formas. Con esta comparación, primero conocían los elementos que iban a formar parte del modelo, teniendo en especial consideración, aquellos que evocarían dicho objeto en la mente de los jugadores. (Robert J. Lang en KAZ: Pushing the Virtual Divide, 2014).

Tras finalizar el proceso de modelado, se lleva a cabo la fase de renderizado. Este proceso consiste en generar la imagen bidimensional del modelo tridimensional mediante un algoritmo que calcula la interacción de las cámaras, luces y materiales con el entorno y el modelo (Blender, 2021). Este proceso de renderizado se puede llevar a cabo creando objetos con un elevado número de polígonos, o, en el caso de los videojuegos, suele optarse por la técnica de modelado de bajo poligonaje (también conocida como *low poly*), en la que los objetos estén creados con pocos polígonos y detalles. Además, se tiene en cuenta que la topología que es la manera en que están distribuidos los vértices, aristas y caras que conforman la malla del objeto en 3D. La topología tiene que ser correcta para evitar problemas posteriores en las fases de

texturizado y animación, debido a deformaciones de malla que pueden ocasionar efectos no deseados (IFP, 2020). Con un cuadrilátero (*quad*) se genera una topología correcta en la que las mallas están ordenadas (tiene una dirección concreta en comparación con los triángulos). Además, estas mallas que componen un cuadrilátero se pueden dividir en triángulos (*tri*). Este proceso se conoce como triangulación. Los videojuegos generan triángulos no solo a partir de cuadriláteros, sino también de otros polígonos. Cuantos más polígonos compongan la malla de un objeto 3D, más largo será su proceso de renderizado, donde sucede el proceso de triangulación. Una correcta topología de la malla compuesta solo de cuadriláteros implica que este proceso solo pueda suceder de dos maneras y no aleatoriamente como ocurre con los polígonos de más lados. Sin embargo, modelado en baja resolución se debe realizar con triángulos, dado que es la forma geométrica con la que trabajan los motores gráficos. De esa manera, se podrá sacar el mayor partido a la calidad de las texturas sin sacrificar la optimización del rendimiento del videojuego. Además, la malla no ha de tener un gran detalle, sino que son las texturas las que se encargarán de esta parte (Tokio School, 2020).

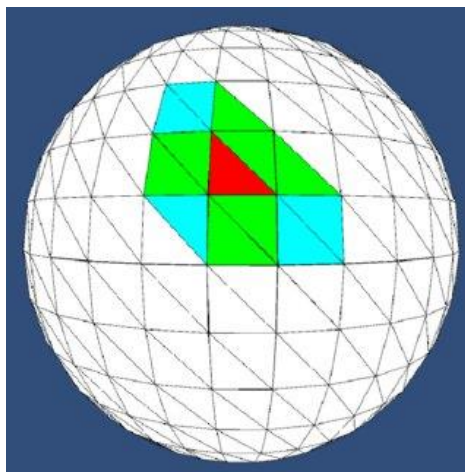


Figura 8. Triangulación de la malla de esfera con topología correcta. Fuente: Vendrell, E. (2018).

Además, en el renderizado influye el LOD (*level of detail* o nivel de detalle). Según la función que vaya a tener un objeto en el videojuego, la complejidad de su modelado y la distancia que tenga respecto a la cámara, tendrá un mayor o menor nivel de detalle. Por ello, los objetos que más lejos se encuentren, pueden tener una menor calidad visual que los más cercanos. Según el jugador se acerque, los objetos irán ganando una mayor calidad en cuanto a sus texturas, de modo que el jugador aprecie más detalles. El jugador no se podrá acercar a ciertos objetos estén muy lejos, por lo que las texturas tienen una menor calidad. De esa manera, no se renderizará

a una calidad óptima los objetos que no se puedan apreciar. Por otro lado, puede haber ciertos objetos cuyas texturas varíen en cuanto a su calidad. Esta variación se realiza según la distancia a la que se encuentre el jugador del objeto: cuanto más lejos esté, de menor calidad será la textura, y, según se acerque, la textura irá ganando calidad. Este nivel de detalle se debe controlar con el fin de que la calidad y rendimiento de la imagen que se desea obtener esté equilibrada. Otro de los factores que condiciona el renderizado de la imagen es la plataforma donde se lanza el videojuego. Cada plataforma tiene unas características que limitan lo que puede renderizar. Por ello, si el tamaño de lo que se quiere visualizar sobrepasa lo que la plataforma puede renderizar, el videojuego no se reproducirá adecuadamente, sino que tomará minutos u horas para generar cualquier simple movimiento (Pluralsight, 2014).

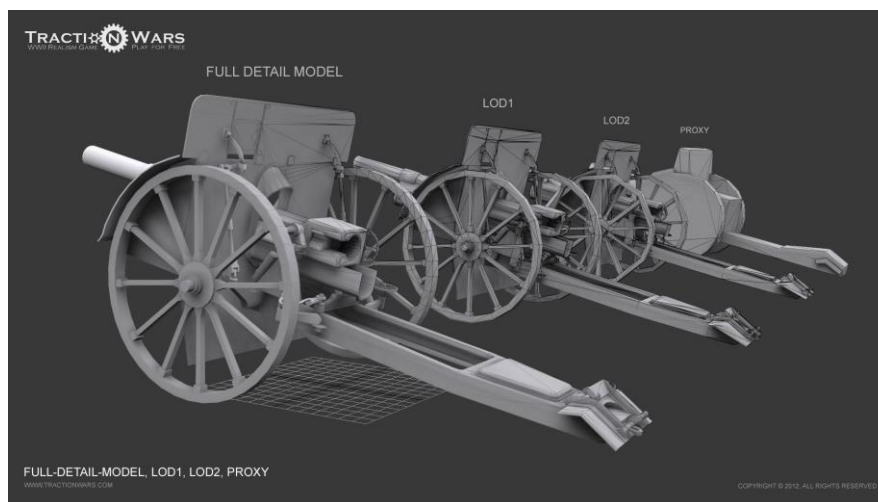


Figura 9. Cuatro *levels of detail* de un objeto según la distancia con respecto al jugador. Fuente: 3djuegos.

No obstante, el diseño de los videojuegos en tres dimensiones ha experimentado un crecimiento progresivo desde los años 70 (V-ART, 2018). Desde esa década, comenzaron a utilizarse los programas de modelado en 3D en el campo del entretenimiento audiovisual. Sin embargo, se intentaba hacer una simulación de las tres dimensiones mediante la utilización de fondos de dos dimensiones. En la década de los 80, surgieron herramientas, como el *3D construction kit*, con las que se comenzaron a desarrollar videojuegos en tres dimensiones, aunque corrieran a una baja velocidad. Algunos ejemplos son *Elite* y *Tau Ceti*. Sin embargo, el primer videojuego que es considerado en tres dimensiones es *Virtua Fighter*, del año 1993. Los personajes estaban modelados con polígonos y podían avanzar, retroceder y moverse de lado a lado. Esto mismo conllevó que los videojuegos comenzaran a diferenciarse entre ser de tres dimensiones o de

dos. Los géneros se convirtieron en 3D fueron los de carreras y los *shooters*. Los videojuegos que surgieron a partir de ese entonces hicieron aprender, tanto a diseñadores y programadores como a jugadores, a moverse en el espacio en tres dimensiones, ya fuera para crear el videojuego o para jugar (Solo, 2018). Con la llegada de los 16 y 32 bits, los videojuegos podían ser de dos o tres dimensiones. A medida que la tecnología avanzaba, los videojuegos evolucionaban, por lo que los requisitos de *hardware* eran mayores. Surgieron programas como *3ds Max* que era un *software* avanzado para el modelado en 3D. Además, los motores de render, encargados de presentar la escena y objetos tridimensionales en dos dimensiones según el LOD y los efectos visuales definidos, mejoraron con la llegada del nuevo siglo, como el caso de *Vray*. Eso permitió obtener una mejor calidad gráfica de texturas, materiales, iluminación y sombras (Vallejo & Martín, 2013).

3.4.2. Texturizado de videojuegos

El texturizado de videojuegos es una de las fases de la producción de un videojuego.

El texturizado es el proceso de asignar y adaptar las texturas, es decir, imágenes, a un polígono o a una malla 3D. La textura envuelve el modelo para dotarlo de características materiales e interactuar con el ambiente (como la iluminación) para que parezca realista y, al formar parte de la experiencia gráfica del videojuego, están definidas por el estilo artístico del diseño. El proceso de texturizado influye en la carga computacional requerida para procesar un videojuego. Esta carga computacional influye en el rendimiento del dispositivo que lo reproduzca y del desarrollo del propio videojuego, por lo que se deben optimizar tanto los procesos como los resultados del texturizado, para adecuarse a las necesidades técnicas de cada proyecto. Con el fin de compensar este condicionante, en los videojuegos se comenzó a ofrecer la imagen prerenderizada por unos equipos más potentes que el de los jugadores. Esto implica que las imágenes no sean renderizadas a tiempo real mientras se juega y que también sean de menor calidad visual (Mac Donald, 1998). En la actualidad, las mejoras de *hardware* y *software* (concretamente los motores gráficos) permiten que las imágenes sean renderizadas en tiempo real.

Un motor gráfico es una de las funcionalidades de un motor de juego (o *game engine*). Los motores de juego son herramientas para desarrollar los videojuegos. A parte de motor gráfico, ofrecen otro tipo de motores como detector de colisiones, simulación de leyes físicas, gestor de sonido, inteligencia artificial y control de interacción (Manrubia, 2014). Con esta división

del contenido, se pueden utilizar aspectos del juego, en vez de crearlos desde cero. Además, se deben tener en cuenta una serie de factores para escoger el motor de juego para desarrollar un videojuego. Estos condicionantes son la codificación, la plataforma de desarrollo, el género, la perspectiva 2D o 3D y las habilidades artísticas (González, 2018). Una de las técnicas que se puede utilizar para optimizar el escenario de un videojuego y su jugabilidad es la comprensión de texturas. Esto consiste en reducir su tamaño para que ocupen menos espacio en el disco de almacenamiento y para que la lectura sea más rápida (Unity Technologies, 2016).

Los videojuegos comenzaron teniendo como textura un color. Con la consola Atari 2600 los gráficos ganaron complejidad y con el procesador de 8 bits de la consola NES de Nintendo, la paleta de colores se amplió a los 48, un mayor número de objetos con los que se podían interactuar (conocidos como *sprites*) aparecían en pantalla; con la consola Master System se lograba mostrar 32 colores en pantalla. Las paletas de colores se ampliaron con los 16 bits de la consola Mega Drive de SEGA. La cantidad de colores que logró mostrar en pantalla fue de 128. Super Nintendo avanzó hasta ofrecer 512 colores (García, 2015). Además, los 32 bits hicieron evolucionar el 3D, con la llegada de PlayStation. Su nacimiento causó que la complejidad gráfica aumentara considerablemente, incluyendo escenarios pre-renderizados, optimizando el rendimiento de las consolas. Más adelante, con la llegada de los 128 bits se consiguieron unos acabados con un mayor detalle, sobre todo en los DVD, aunque no llegaban a la calidad de las películas de animación de principios del siglo XXI. Es en ese momento cuando PlayStation y Microsoft comienzan una lucha por la creación de la consola más potente que permitiera ofrecer unos mejores gráficos, Nintendo se centró en la jugabilidad (Muy Computer, 2019).



Figura 10. *Street Fighter II: The world warrior* en Super Nintendo (1992) y PlayStation (1998). Fuente: VCDECIDE.

3.5. Software de modelado y texturizado 3D

En la actualidad existen diversos *software* para modelar y texturizar en tres dimensiones. Algunos de estos programas son SketchUp que se centra en la creación de entornos arquitectónicos y diseño industrial, al igual que 3ds Max y Maya que también incluyen opciones avanzadas de texturizado y animación y Blender, un programa de código abierto y gratuito que se centra en animaciones y efectos visuales (Educba, s.f.).

El *software* de modelado y texturizado 3D pone a disposición de los usuarios diversas herramientas con las que irán formando mallas, vértices, aristas y caras. Estos elementos pueden crearse a través de formas primitivas, curvas o incluso vértices, dependiendo de la técnica de modelado que se prefiera según la forma. No obstante, ninguno de estos procedimientos de trabajo puede prescindir de un modelo para poder reproducirlo. Por ello, una de las ventajas que este *software* tiene es que se puede incrustar una imagen en 2D que servirá de plantilla para el modelado en tres dimensiones.

Para llevar a cabo el modelado poligonal, se parte de una serie de mallas primitivas que ciertos *software* de modelado incluyen por defecto como el plano, el cubo, el cilindro y la esfera. Estas figuras se van modificando en función del modelo que se vaya a crear. Este modo de trabajo suele ser el más sencillo de todos y, para llevarlo a cabo, el cubo es la figura que resulta más versátil, pues con las herramientas y los modificadores del software se puede obtener casi cualquier forma. No obstante, en vez de utilizar una figura primitiva se puede optar por partir de un solo vértice. Con él seleccionado se irán creando los vértices sucesivamente mientras se va creando el contorno de la malla base sobre la que se trabajará más adelante pasando de ser una malla plana a un objeto con volumen en sus tres dimensiones (Blender, 2021).

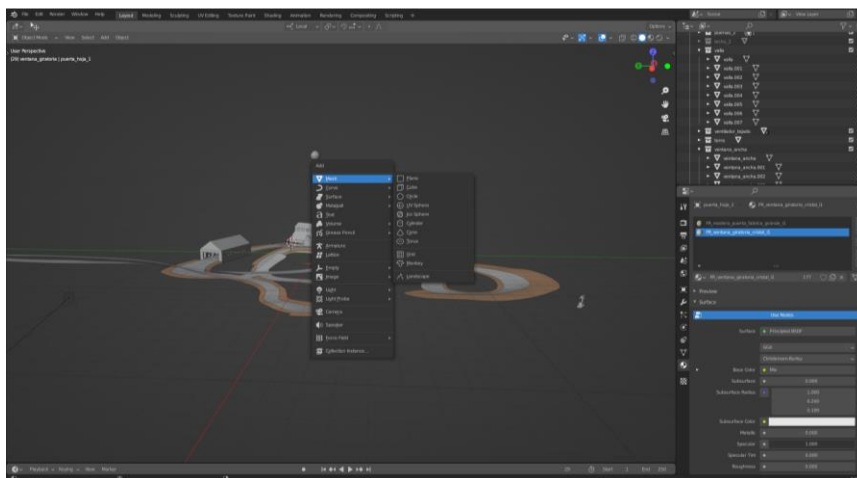


Figura 11. Pantalla de trabajo en Blender y mallas primitivas.

También se puede llevar a cabo la técnica de modelado mediante curvas. Resulta ser más complejo que los anteriores, dado que ofrece una mayor libertad, aunque el ahorro de vértices es enorme. Los tipos de curvas más comunes con las que se suelen trabajar son *Bézier* y *Nurbs*. Para poder editar las curvas, se modifica la localización y rotación de sus nodos que indican la forma que esta va a tener. Las curvas *Bézier* son más sencillas, pues simplemente hay que editar sus nodos y vectores para obtener el resultado deseado. Luego, solo es necesario convertirla en una malla para poder trabajar sobre ella como un objeto. No obstante, este tipo de curvas no puede representar las curvas cónicas (Flores, 2010).

Las curvas de tipo *Nurbs* son muy sencillas de procesar mediante ordenador, dado que no suelen tener errores en cuanto a su representación y no necesitan de mucha memoria para procesar el cálculo matemático. Aunque se basan en las curvas de tipo *Bézier*, las *Nurbs* se definen por cuatro elementos fundamentales (y no dos como en el caso de las otras). Estos parámetros son los puntos de control (aproximación de una curva), pesos (atracción de una curva según la distancia a su punto de control), vector de nodos (los nodos de las curvas de tipo *Nurbs* no están distribuidos de manera uniforme como en el caso de las *Bézier*) y grado (indica la suavidad de la curva en función de si este valor es más o menos alto). La gran ventaja que ofrecen es la facilidad con la que controlar la suavidad de las curvas, evitando que se tuerza (Flores, 2010).

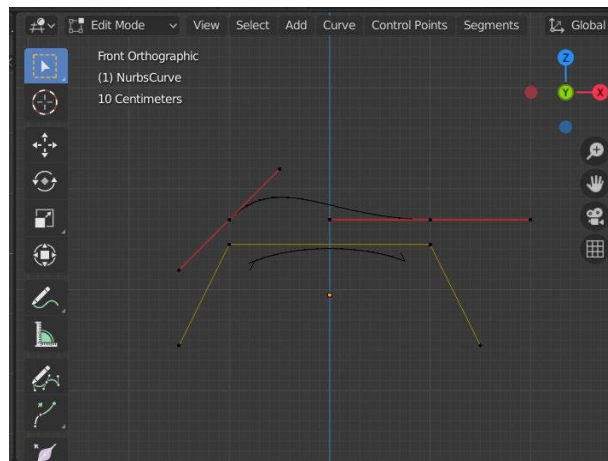


Figura 12. Curva *Bézier* (arriba) y *Nurbs* (abajo) en Blender.

Para poder realizar el modelado, existen una serie de controles básicos a la hora de diseñar en tres dimensiones. Estos son el escalado, para aumentar o disminuir el tamaño, la rotación y la localización. Todos ellos funcionan sobre uno o varios objetos, aristas, vértices y caras en los ejes X, Y, Z y en función del resultado que se desea obtener. De carácter algo más avanzado,

para conseguir efectos más trabajados, los elementos mencionados anteriormente se pueden subdividir, de modo que la malla pueda ser trabajada en más puntos, logrando una mayor calidad y detalle. También se pueden suavizar los vértices y así eliminar la tosquedad del modelo creado e incluso se pueden realizar extrusiones, es decir, extracción del elemento seleccionado para poder crear formas más complejas. Además, los *software* ofrecen modificadores, es decir, operaciones automáticas que actúan sobre la geometría de un objeto para transformarlo de alguna manera y generar efectos avanzados.

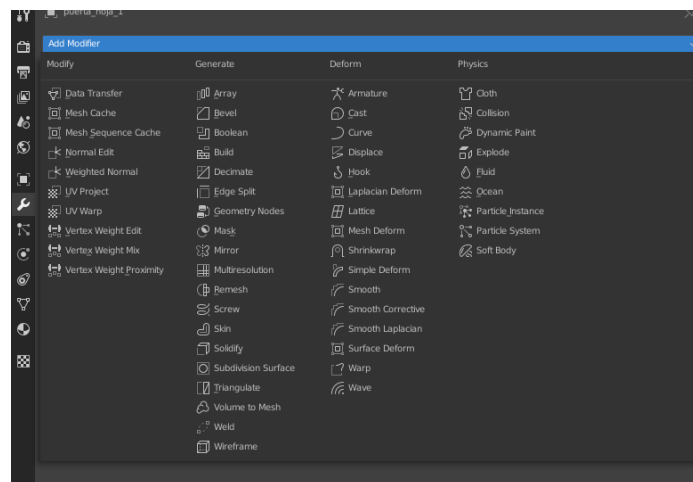


Figura 13. Panel de modificadores de una malla en Blender.

También, ciertos *software* como Blender incluyen herramientas más intuitivas para llevar a cabo la técnica de modelado de escultura digital. Con diversas herramientas se puede deformar la malla y esculpir la forma deseada. Este modo de trabajo es más frecuente utilizarlo a la hora de crear personajes y no tanto cuando se quieren generar otras construcciones industriales. Después de realizar el modelado, se puede llevar a cabo una retopología. Este proceso volver a crear el objeto tomando como referencia el que ya había sido obtenido. Esta fase se realiza para reducir el número de polígonos y que estos sean específicamente cuadriláteros (*quads*), evitando cualquier triángulo en la malla que pueda provocar deformaciones posteriores (Trazos, 2021).

Una vez finalizado todo el proceso de modelado y de retopología en caso de que fuera necesario, se texturiza. Este proceso se puede llevar a cabo a modo de textura imagen o *bitmap*, o bien, con el tipo procedural. La principal diferencia entre los tipos *bitmap* y procedural reside en que el primero utiliza una imagen que se incorpora sobre la malla (proceso de mapeado o

mapping) y el procedural funciona mediante algoritmos de un *software* como Blender. La imagen aportará más realismo que el cálculo matemático y, por tanto, un mejor acabado dados los efectos que se pueden incorporar, pero el tamaño del archivo aumentará en comparación al archivo donde se usaron texturas procedurales (Lobo, 2000).

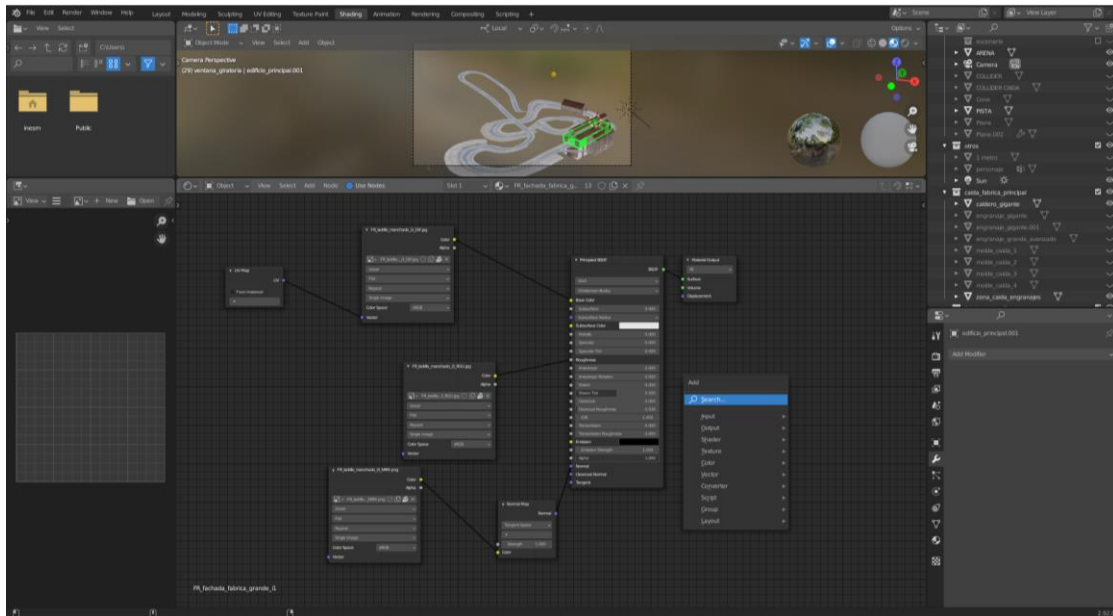


Figura 14. Shading en Blender con una textura bitmap aplicada.

La resolución de la textura es la cantidad de píxeles de una imagen. Para que el renderizado de las mismas sea más rápido las medidas en píxeles de altura y anchura de una imagen tienen que ser potencia de dos como, por ejemplo, 512x512, 1024x1024, 2048x2048. Con una mayor resolución, más nitidez tendrá la textura, pero ocupará un mayor espacio en la memoria del dispositivo (Lobo, 2000). Para optimizar la asignación de texturas, estas tienen que ser cuadradas, es decir, que tengan el mismo número de píxeles tanto de alto como de ancho. De esa manera, se procesarán más rápidamente. Además, las texturas imagen pueden repetirse en el caso de que sean del tipo *tileable* o mosaico. Al unir dos o más texturas de este tipo, su costura o corte no es apreciable. De esa manera, se puede utilizar una textura más de una vez, ofreciendo una mayor calidad al objeto (Kaur, 2003).

Para poder aplicar correctamente una textura imagen a un modelo, se debe desenvolver la malla del objeto. A este proceso se le conoce como mapeado UV. Para ello, primero se seleccionan una serie de aristas para crear una costura que será la referencia que guiará la desenvoltura de la malla. Cada uno de los trozos resultantes de dividir la malla tras desenvolverla se llama “isla”. Las islas se pueden ajustar modificando la posición, escala y rotación de cada uno de

sus vértices, aristas o caras para generar efectos distintos al texturizar y para que no se superpongan las unas a las otras evitando que se genere una repetición de la textura.

En los videojuegos se utiliza una técnica de renderizado de texturas a tiempo real conocida como *PBR* (*Physically Based Rendering*). Es un modelo creado por Disney en el que se basa *Principled BSDF* de Blender para aplicar y combinar mapas de texturas de distintos atributos al mismo objeto (Blender, 2021). Además, este método de renderizado ofrece resultados realistas, ya que su algoritmo se basa en funciones precisas a la física y responden correctamente en todas las condiciones de iluminación (McDermott, 2018). La cantidad de mapas de texturas a incluir dependerá del objeto y de su material.

Algunos de los mapas de texturas más comunes son (McDermott, 2018):

- *Base color* que es el color del material. También se le conoce como mapa albedo o difuso.
- *Specular* que indica cuánta luz se refleja sobre un objeto.
- *Roughness* el cual se trata del mapa de rugosidad. Este mapa da información sobre las irregularidades de la superficie que provocan que la difusión de la luz y los cambios de dirección de la misma. Se utiliza para dar detalles como arañazos, desgaste y manchas. Este mapa se representa mediante zonas blancas que representan las áreas rugosas y zonas oscuras para indicar reflexión.
- *Normal* que representa el relieve de un objeto mediante la creación de sombras como simulación del modelado en esa superficie con el fin de consumir menos recursos.
- *Height* es el mapa de altura y se combina con el mapa *normal* para generar más detalles en cuanto a los golpes, rasguños y manchas que se simulan en la superficie del objeto.

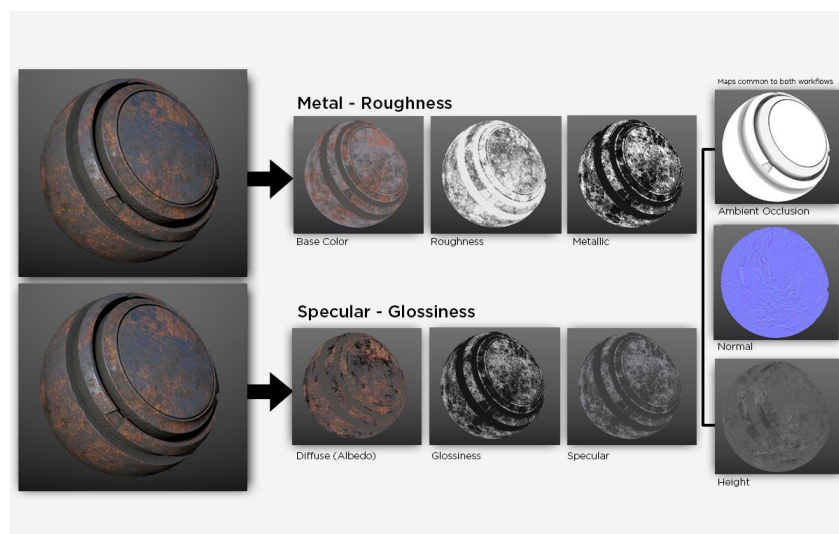


Figura 15. PBR y mapas de textura para un objeto metálico. Fuente: Substance Academy.

Cada uno de los mapas de texturas se irán integrando al material creado y asignado a un objeto para darle las propiedades que tiene el material en la realidad.

Finalmente, con el fin de llevar a cabo el renderizado, se debe incluir iluminación y una cámara. Además, se debe seleccionar el motor de renderizado que ofrecerá resultados diversos en cuanto a la calidad de la imagen y al efecto más o menos realistas.

El *software* escogido para llevar a cabo las fases de modelado y texturizado de este proyecto es Blender, tal y como se decidió en la fase de selección de *software* durante la preproducción de este proyecto. Blender proporciona figuras primitivas sobre las que se pueden aplicar modificadores. Algunos de ellos resultan de enorme utilidad para crear más caras (modificador *Subdivision surface*) o a la hora de modelar la mitad de un objeto (modificador *Mirror*), reduciendo la carga de trabajo. Además, en el modo *sculpting* se puede llevar a cabo el modelado de escultura digital para crear objetos o para editar los ya existentes.

En Blender se lleva a cabo el proceso de texturizado en la ventana de *Shading* donde se pueden editar los nodos para introducir propiedades y efectos avanzados bien sea con otras imágenes o con algoritmos. El mapeado UV se realiza en la pestaña de *UV editing*. Para desenvolver la malla, Blender ofrece una serie de opciones para realizar este proceso rápidamente y obtener distintos resultados al desenvolver la malla. Esto se puede realizar de manera inteligente mediante el *Smart project*. Además, se pueden editar texturas pintando directamente sobre el modelo 3D o sobre la textura en *Texture Paint*. Cada uno de los mapas de texturas se irán uniendo mediante los nodos de la pestaña de *Shading* en Blender para atribuir propiedades al material del objeto y obtener resultados realistas en el renderizado.

Además, para renderizar la imagen, Blender ofrece tres motores de renderizado. Uno es *Workbench*, el cual es a tiempo real y pensado para ir modelando mientras se ve una imagen renderizado, *Eevee* está basado en el renderizado a tiempo real y es rápido, aunque sus resultados no sean tan buenos como los de *Cycles*, el cual ofrece los resultados más realistas, pero tarda más en renderizar dado que es de tipo *ray tracing*, es decir, calcula la ruta de cada rayo que proviene de la cámara de la escena.

4. Metodología

En este apartado se exponen las fases de trabajo y las herramientas que se emplearán posteriormente en el desarrollo del proyecto.

4.1. Preproducción

La preproducción es la primera fase dedicada a investigar, estudiar y analizar todo el ambiente alrededor del proyecto y de los objetivos que se quieren desarrollar. Una buena planificación antes de trabajar hará que el desarrollo sea más eficaz y eficiente. La preproducción incluye la documentación y la planificación desarrollada en un esquema de etapas y tareas que se incluye en el “Anexo 1” y el “Anexo 2”.

4.1.1. Documentación

El proceso de documentación consiste en una investigación previa sobre la época en la que se ambienta el videojuego. Se recurrirá a archivos fotográficos, artículos y libros *online* para definir el proceso de industrialización y funcionamiento de fábricas de los siglos XIX y XX. De esa manera, se recopilará la información e imágenes suficientes para crear los elementos necesarios que ambienten el entorno del videojuego de acuerdo a la época. Tal y como se muestra en el apartado de “Marco teórico”, la época en la que se ambienta el videojuego es durante la Segunda Revolución Industrial, siendo característicos el hierro, el acero y el hormigón. Por ello, la documentación se centrará en la obtención y transformación tanto del hierro como del acero.

4.1.1.1. El circuito

En primer lugar, se partirá del esquema del circuito que ya estaba modelado por el equipo de trabajo para realizar una serie de adaptaciones del mismo. De esa manera se conseguirán unificar los personajes y vehículos a la pista ya creada.

Para llevar a cabo dicho trabajo, se buscarán referentes, dando preferencia a los que ya se establecieron anteriormente. *Mario Kart* fue el videojuego de carreras más señalado como referencia, debido a su bajo poligonaje y por su estilo desenfadado y colorido. Aunque se parta

de la realidad para rediseñar la pista, siempre se tendrá en cuenta la originalidad y la creatividad con el fin de que el circuito resulte más divertido para los jugadores.

4.1.1.2. El escenario

Antes de comenzar a llevar a cabo el proceso de modelado, es necesario realizar una investigación previa que permita establecer los elementos que compondrán el diseño del escenario del circuito.

Una vez conocidas la ambientación y los objetivos del videojuego, es el momento de realizar la tarea de investigar el estilo artístico de cada elemento en base a la época en la que está ambientado. Para ello se consultarán en su mayoría registro de fotografías y otra serie de documentos que den como pista el funcionamiento de los elementos industriales que se quieren incorporar al diseño.

Asimismo, se consultarán las referencias que se habían fijado para el circuito como fuente de inspiración para incorporarlos como elementos creativos en el escenario. Estas referencias serán en su mayor parte imágenes de archivos fotográficos *online* que sean acordes a la época, que formen parte del proceso de funcionamiento de una fábrica y que sean elementos simples, de manera que se puedan modelar con pocos polígonos y sean reconocibles.

4.1.1.3. Las texturas

El estilo de los elementos, en concreto su texturizado, será definido por el equipo de trabajo ÍTACA. Por tanto, el texturizado se realizará en base a los requisitos que se han fijado para obtener un resultado coherente y unificado por todos los miembros del equipo.

Las texturas se obtendrán en su mayor parte de bancos de las mismas en páginas web como *Sketchup Texture Club* o *FreePBR*, dado que ofrecen texturas que se pueden repetir sin que se aprecien los límites de principio y/o final de la textura (también se conocen como texturas tileables). Además, se tendrán en cuenta las dimensiones optimizadas para videojuegos antes expuestas en el “Marco teórico”.

4.1.2. Selección del software

Cuando se va a realizar un proyecto es necesario planificar con qué herramientas se va a trabajar. Esta tarea gana importancia cuando el trabajo se va a realizar en grupo para obtener los mismos resultados siguiendo las líneas de diseño del proyecto. Por ello, el equipo de trabajo establecerá con qué programas y sus respectivas versiones se va a llevar a cabo el desarrollo del proyecto. Este flujo de trabajo es el *pipeline*, que es, según Dunlop “la expresión técnica de una compleja red de actividades humanas” (Dunlop, 2014: 143).

Para realizar la fase de modelado y la de texturizado, se escogió Blender en sus versiones 2.91.2 y 2.92, dado que es el programa con el que está desarrollado el trabajo previo de este videojuego. Además, aunque la versión sea superior al del resto del trabajo, soporta las versiones anteriores. También, se ha escogido porque es el que se utiliza en el Grado de Comunicación Audiovisual y en el Máster en Comunicación y Desarrollo Multimedia y sobre el que se tienen conocimientos más avanzados. Además, tiene todas las herramientas y funciones necesarias para el llevar a cabo este proyecto, tal y como se especifica en el “Marco teórico”.

Para realizar pruebas rápidas sobre el modelado del circuito, se utilizará el programa Unreal Engine en el que importará dicho archivo y tomará como referencia para hacer los ajustes pertinentes de cara a integrarlo lo más definitivo posible en la programación del videojuego.

4.1.3. Planificación

Tras la documentación, es necesario saber qué pasos tomar antes y después siempre teniendo presentes los objetivos. La planificación de estas tareas se llevará a cabo intentando prever las dificultades e inconvenientes que se vayan a tener durante el desarrollo, con el fin de ahorrar tiempo en la ejecución del trabajo y para impedir las desviaciones de los objetivos previamente establecidos. Además, una planificación correctamente desarrollada hará que el proyecto se comprenda mejor y que la producción del mismo sea más eficiente (Gómez, 2016). En primer lugar, se partirá de lo que ya estaba fijado como base del videojuego por el resto del equipo de trabajo. También se tendrá en cuenta las limitaciones en cuanto a programación y al tiempo de realización del proyecto.

Una vez realizada dicha investigación y puesta en común con los miembros del equipo, se pasará a la fase de establecer patrones de trabajo y repartición de tareas de diseño. Sin embargo, la planificación no será determinante, pues el desarrollo del trabajo puede variar en función de

las dificultades y modificaciones en cuanto al diseño para que el videojuego esté optimizado (Salen & Zimmerman, 2003).

4.1.4. Trabajo en equipo

Este Trabajo Fin de Máster se realizará paralelamente al trabajo de un equipo que se dedicará a desarrollar otras partes del proyecto de la creación del videojuego *Fastastic Roads*. Los miembros del equipo tienen que estar coordinados para poder llevar a cabo el proyecto de manera satisfactoria. No solo tiene que existir comunicación para establecer patrones de trabajo, sino durante el desarrollo del proyecto se deben corregir los diseños con el fin de mejorarlos obteniendo distintos puntos de vista o incluso idear otros nuevos.

Con una buena comunicación y coordinación, el proyecto irá fluyendo, haciendo que cada una de las partes que lo conforman, bien sea diseño o bien programación, puedan ir avanzando en cada una de las tareas que les corresponde para lograr el objetivo de crear el videojuego (Gilmore & Moraes, 2002).

4.2. Producción

La etapa de producción consistirá en el diseño y creación de elementos que compondrán el escenario del videojuego, concretamente estos objetos se modelarán en 3D y se les texturizarán. Esta fase seguirá lo que se ha planificado en la etapa anterior: se usarán las herramientas que se han planificado y se seguirá el procedimiento de trabajo previsto (Miriadax, 2019).

4.2.1. Diseño

Para realizar el diseño del circuito se tomará como referencia principal el que ya está creado por el equipo de diseño. Además, se recurrirá a archivos de imágenes sobre la época histórica en la que está ambientado y a otros videojuegos de carreras para establecer las dimensiones y peralte de la pista. Para llevar a cabo el diseño, se utilizará Adobe Photoshop con la herramienta del pincel para bocetar y señalar las subidas y bajadas de la pista. Además, se recurrirá al resto del equipo para poder hacer las modificaciones pertinentes teniendo en cuenta las opiniones

con respecto a las limitaciones técnicas, creatividad, etc. De esa manera, se conseguirá un diseño definitivo sobre el que comenzar a desarrollar el proyecto.

En cuanto al diseño del escenario, también se tendrá como referencia principal los elementos que ya se habían dispuesto en el primer boceto por el equipo de diseño, así como otras ideas que hayan ido surgiendo. Siempre se tendrá en cuenta la ambientación histórica del videojuego, y concretamente la fabricación del hierro, al igual que otras referencias de videojuegos que sirvan de inspiración para que el escenario resulte más divertido. Al igual que el circuito, el diseño del escenario se realizará en Adobe Photoshop, además de Blender, programa en el que se situarán los elementos de manera muy básica para poder trabajar sobre el diseño de los elementos desde el punto de vista que tendrán los vehículos.

4.2.2. Modelado

En esta fase se desarrollará el proyecto teniendo en cuenta la planificación y los diseños previamente realizados. Para desarrollar esta etapa cada uno de los objetos se modelarán en archivos independientes, con el fin de facilitar el mismo proceso de modelado, el rendimiento del ordenador y la exportación posterior. El trabajo de modelado se realizará partiendo de un diseño previo sujeto a cambios sobre el que se distribuirán los elementos según la vista que los jugadores tengan desde el vehículo. Luego, según se vayan creando los objetos en tres dimensiones, se irán realizando pruebas para que estos elementos encajen mejor en el diseño y se irán llevando a cabo modificaciones con el fin de generar adaptaciones óptimas.

El tipo de modelado que se llevará a cabo será el poligonal, pues permite crear objetos partiendo de figuras primitivas que son de bajo poligonaje como un cubo o un cilindro de ocho caras. Según el elemento lo requiera, se añadirán más figuras primitivas o se irán editando los vértices, aristas y caras mediante la extrusión, aumento o disminución de la escala, rotación y desplazamiento. Además, se utilizarán modificadores sencillos que permitan reducir el número de polígonos que forman el objeto y que estos sean cuadriláteros.

4.2.3. Texturizado

Siguiendo la planificación, el proceso de trabajo comenzará estableciendo el nivel de detalle de cada uno de los objetos según la función que vayan a tener y según la distancia a la que se encuentre el jugador.

Además, se tendrán en cuenta las características de cada material y los resultados que se quieren obtener para realizar los distintos mapas de texturas de *base color*, *normal*, *roughness* y *specular*. Todas ellas serán fotografías cuya resolución variará en función de la distancia que ese elemento del escenario vaya a tener con respecto al jugador.

Con el fin de que el rendimiento del videojuego sea el óptimo, todas las texturas tendrán relación de aspecto 1, es decir, que serán cuadradas. Además, las dimensiones de las texturas serán potencia de dos para que sean eficientes y no necesiten reescalarsen en el renderizado de la imagen (Unity Documentation, 2016). El tamaño de cada textura variará en función de la distancia de la cámara con relación al objeto. Lo que más cerca se encuentre un centímetro de la realidad corresponderá con 4 píxeles, para lo que esté lejos un centímetro serán 10 píxeles y lo que vaya a estar en cinemáticas o primeros planos un centímetro será un píxel. De esa manera, las texturas tendrán la calidad suficiente sin que afecte negativamente al rendimiento del videojuego. Por tanto, el proceso de trabajo será diferente según la complejidad del objeto y del nivel de detalle que se quiera conseguir. Para los elementos que sean más complejos y que necesiten tener un mayor detalle, se llevará a cabo una desenvoltura de su malla y se editará en Adobe Photoshop CC, para luego exportar el resultado a Blender donde únicamente se le asignará dicha textura al objeto. Para los objetos sin tanto nivel de detalle y menos complejos, se importará la textura y en Blender se modificará la posición, escala y rotación de las islas, tras desenvolver la malla, según el efecto que se quiera conseguir.

Los materiales que se obtendrán serán piedra, revestimiento de paredes que sean antiguas y desgastadas, con aspecto industrial; y metales, concretamente, hierro, acero y óxido. Estas texturas no se utilizarán exclusivamente sino que se reutilizarán para otros objetos y entre todo el equipo de trabajo. De ese modo, el estilo visual estará unificado y será coherente. Además, se crearán los mapas de texturas necesarios para cada material y se realizarán modificaciones para obtener los resultados óptimos. El equipo ÍTACA estableció también en qué formato se encontraría cada mapa de textura. De ese modo, los mapas de textura *albedo*, *roughness*, *specular* y *height* estarían en JPG y el mapa de textura *normal* en PNG.

Finalmente, para que exista unidad visual y originalidad, se llevará a cabo un proceso de postproducción con cada mapa de textura para aplicar efectos originales a los materiales de los elementos que componen el escenario. Este proceso de postproducción se llevará a cabo en *Deep Dream Generator* que es una herramienta web, cuyo funcionamiento se basa en algoritmos de inteligencia artificial y fusiona dos imágenes. Esta herramienta se utilizará con

las texturas albedo seleccionadas por el equipo de trabajo y otra imagen que genere efectos originales y acordes al estilo del videojuego.

4.2.4. Renderizado

Esta fase de renderizado en este proyecto no será necesaria, dado que no se exportará como un contenido de imagen o vídeo. Para implementar esta parte del escenario en el videojuego completo, la exportación se realizará en otro formato que el ordenador sea capaz de interpretar para establecer acciones de programación sobre él y que funcione correctamente.

Sin embargo, se realizarán renderizados puntuales para comprobar que la visualización de los elementos sea la correcta antes de incorporarlos en el proyecto completo. Por ello, las características de la iluminación serán similares a las que tendrá el videojuego. De esa manera, se verificará la cantidad de zonas de entrada de luz como ventanas, la distribución de objetos y el color de los mismos. Además, la cámara se colocará en puntos concretos en los que el jugador vaya a tener una mayor visión de ciertas zonas del escenario con el fin de aumentar la calidad visual en ellas.

5. Desarrollo

Este apartado explica de manera detallada el trabajo que se ha llevado a cabo para alcanzar los objetivos propuestos previamente a la ejecución.

5.1. Preproducción

La fase de preproducción incluye la documentación sobre la época y búsqueda de referencias más específica para desarrollar de manera eficiente la producción del proyecto. Además, incluye la planificación y selección de herramientas que se vayan a utilizar teniendo en cuenta las habilidades personales y las preferencias del equipo de trabajo.

5.1.1. Referencias para el diseño y modelado del escenario

Las búsqueda de referencias e información son necesarias para llevar a cabo la producción del proyecto. A través de Internet, libros y videojuegos se consiguió obtener las fuentes tanto de inspiración como para utilizarlas como base para la etapa de modelado.

En primer lugar, para establecer lo que se iba a modelar se recurrió a páginas y documentos *online*. También, se tuvo en cuenta el trabajo anterior realizado por ÍTACA en cuestiones del videojuego.

Para el diseño del escenario y modelado de los objetos, se recurrieron a archivos fotográficos y libros online donde se encontraron imágenes de la maquinaria para luego reproducirlas en la fase del modelado. Concretamente, fue el libro “Précis de métallurgie” de Pecheux (1910) el que principalmente sirvió para obtener información visual y escrita de las máquinas que componen el escenario de la fábrica. Además, para el modelado *low poly* y otros detalles estéticos fueron los videojuegos *Mario Kart* y *Bioshock*. El primero de ellos fue referencia para el modelado debido a su simplicidad y de manera funcional por algunos elementos creativos que interactúan en el escenario. El primer videojuego de la saga de *Bioshock* sirvió como inspiración estética en cuanto al color y elementos decorativos del escenario que también está basado en la época de la Revolución Industrial.



Figura 16. *Mario Kart 8*. Fuente: Guías Nintendo.



Figura 17. *Byoshock Infinite*. Fuente: HobbyConsolas.

Con el fin de representar la narración del proceso del hierro para fabricar piezas y obtener acero, se tomaron como referencia libros acerca de la fabricación de piezas metálicas, fundición del hierro y obtención del acero de la época en la que se ambienta el videojuego. Sin embargo, al tratarse de un videojuego que tiene que resultar divertido, el componente creativo está presente en cuanto al modelado de los objetos, debido a que su tamaño y estructura a veces no son los reales.

El proceso comienza en la montaña de donde se extraen las rocas, las cuales son transportadas al primer edificio en el sentido de la marcha. Es allí donde las rocas se trocean con unos martillos. Posteriormente, el resultado se lleva a la fábrica más grande donde tienen lugar el fundido y la fabricación de piezas. La fundición del metal se lleva a cabo en los hornos de la planta baja de la fábrica para luego darles la forma deseada con la maquinaria. En la laminadora o perfiladora se da forma al hierro estirándolo; con el martillo pilón se moldea el metal calentado al rojo vivo o en frío a través de un golpe seco. Para fabricar piezas más concretas, se encuentran el torno y la fresadora, ambos para mecanizar piezas, realizar taladros, ranuras y otras acciones más avanzadas. La mortajadora sirve para contornear y tallar principalmente, la

cepilladora para alisar superficies, el taladro para realizar perforaciones y para cortar y plegar chapas se encuentran la cizalla y plegadora, respectivamente. Además, para obtener acero, se encuentra el convertidor de Bessemer, el cual tiene un proceso de fundición más agresivo que el anterior. Después tiene lugar el proceso de la colada que consiste en depositar el resultado de la fundición del convertidor en unos moldes.

La selección de las máquinas que modelar se realizó teniendo en cuenta tanto la funcionalidad de la misma como la época de utilización de la máquina y su forma. El factor de la funcionalidad se debe a que se perseguía una coherencia narrativa en el escenario y entre los elementos de la fábrica acordes a la época en la que se basa el videojuego. Además, las máquinas que se buscaban tenían un número elevado de piezas. Esto no se consideró un problema para el modelado posterior, debido a que se pretendía que las máquinas tuvieran volumen y no fueran planas, con el fin de hacerlas reconocibles en cuanto a que eran máquinas para trabajar metales. No obstante, una de las dificultades que se consideró fue que las representaciones de las mismas en las fuentes consultadas solo se las representaba en un ángulo. Esto implica que el resto de la máquina que no aparece dibujada en las referencias se tenga que imaginar de una manera lógica para que la máquina tenga un sentido de funcionamiento. Además, la colocación de la misma en el escenario influye en lo que el jugador percibirá de ella, así que se consideró la rotación de estos objetos en su localización en el escenario como factor influyente en el diseño del interior de las fábricas.

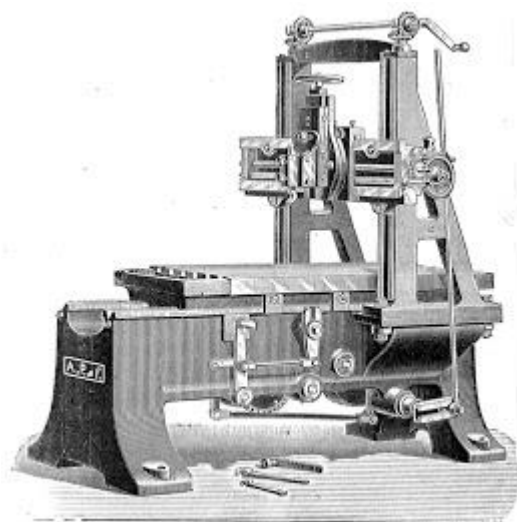


Fig. 29. — Machine à rubeler (Plat).

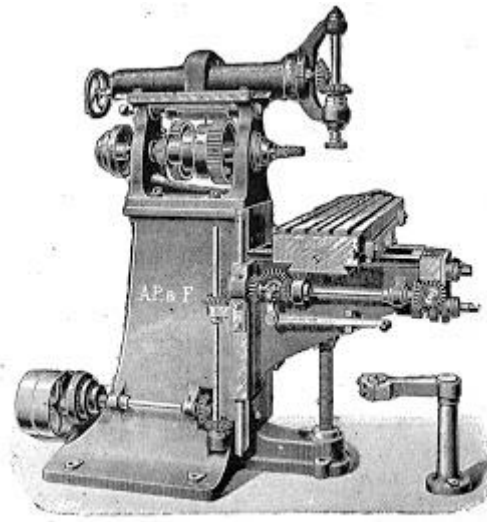


Fig. 30. — Machine à aléser (Plat).

Figura 18. Grabados de una mortajadora y una fresadora del libro “Précis de métallurgie” de Pecheux (1910).

Estas referencias junto a lo planificado junto al equipo de trabajo ÍTACA fueron las fuentes para diseñar las fábricas y la distribución de sus elementos.

5.1.2. Referencias para el texturizado del escenario

La principal referencia que se tuvo para realizar el texturizado del escenario fue la realidad. Siguiendo las fuentes consultadas durante la etapa de documentación para crear los modelos en 3D, los objetos se fueron clasificando en función de los materiales en los que estaban contruidos. Además, también se incluyó la creatividad para atribuir al videojuego un aspecto no tan fiel a la realidad, sino más bien desenfadado.

Para llevar a cabo efectos en la etapa de postproducción de las texturas se recurrió al videojuego *Overwatch* cuyas texturas son simples, con escaso detalle, pero fieles a la realidad. Otra de las fuentes de inspiración fueron los grabados de época, los sellos y planos de ingeniería, los cuales proponen estilos originales y que encajan con la época y el ámbito en los que se basa el videojuego. Con estas imágenes se realizaron pruebas en mapas de textura de objetos ya modelados con la herramienta web *Deep Dream Generator*. Sin embargo, no ofrecían unos resultados que se consideraran acordes a los objetivos perseguidos con el estilo visual del videojuego. Por esa razón, se adaptó el método de trabajo de manera que primero se texturizarían todos los elementos del escenario y después se buscaría un efecto de textura original que encajara en el videojuego y que no resultara excesivamente recargado para evitar confundir al jugador durante su recorrido por el circuito.

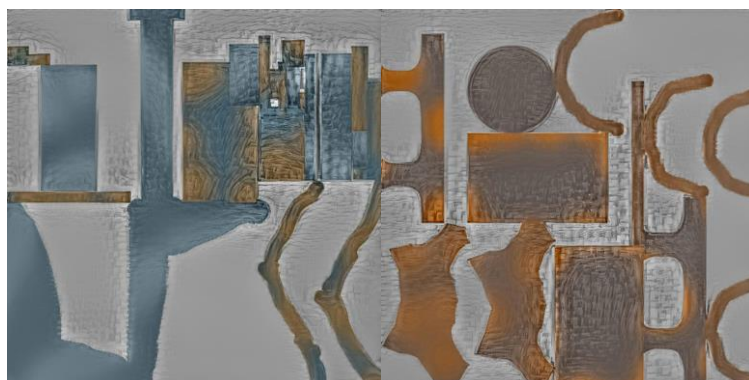


Figura 19. Ejemplos de mapas de textura procesados en *Deep Dream Generator* de una columna.

Estas referencias influenciaron la creación de un estilo visual unificado y original al videojuego, aunque siempre primando la creatividad por todos los miembros del equipo para desarrollar texturas únicas y reconocibles.

5.2. Producción

En este apartado se expone el modo en que se ha desarrollado el proyecto siguiendo la preproducción en cuanto a la planificación y referencias encontradas, además de los problemas encontrados y las acciones que han permitido superarlos para obtener los óptimos resultados.

5.2.1. Diseño del escenario

Antes de comenzar a modelar, se realizaron una serie de diseños que sirvieran como guía del modelado posterior y, de esa manera, que el flujo de trabajo durante esa etapa resultara de una menor complejidad.

Partiendo del boceto del grupo de trabajo ÍTACA y de las referencias encontradas durante la etapa de documentación, se llevó a cabo un boceto de la distribución de la pista en Adobe Photoshop CC, tal y como puede verse en el “Anexo 3”. Después de presentarlo al equipo, se realizó un diseño más avanzado del interior de los edificios que resultara orientativo para el modelado, tanto desde una perspectiva amplia como desde el punto de vista que tendría el jugador en zonas específicas de la pista (véase al “Anexo 3”). Se realizaron bocetos del edificio principal que se iba a modelar, pues es el más complejo y el que más tiempo iba a llevar. Tanto la fábrica más pequeña como la caldera se plantearon como construcciones complementarias que tuvieran sentido narrativo a todo el proceso de obtención y transformación del hierro y el acero. El diseño realizado recoge en buena parte las propuestas realizadas por el equipo de trabajo y recomendaciones sobre videojuegos de carreras, concretamente *Mario Kart*. Al diseño original se añadieron más rectas y curvas de distinta amplitud para aumentar la dificultad del circuito. Además, se introdujo un salto rompiendo una de las zonas y se seleccionaron tramos donde el jugador podría caerse, atajar o salirse de la pista. También se señalaron los tramos en los que habría barreras y se añadieron detalles acerca de cómo iban a ser los terrenos y su interacción con el jugador y otros elementos de carácter decorativo. Sin embargo, el equipo ÍTACA simplificó el diseño para que resultara más sencillo de programar y componer en las primeras versiones del videojuego, aunque se mantuvieron ciertos elementos creativos como los terrenos y el salto.

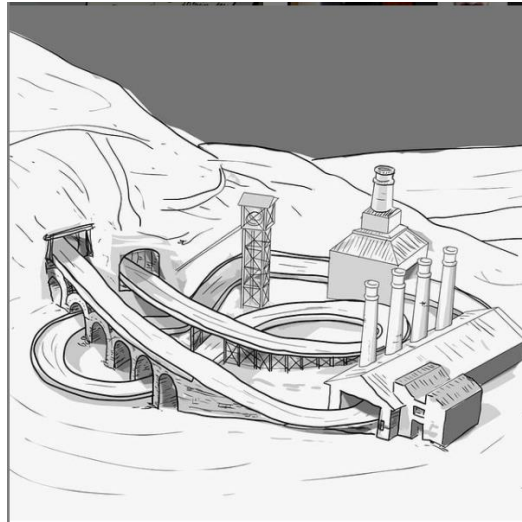


Figura 20. Boceto del escenario por ÍTACA. Fuente: Instagram de ÍTACA (@itaca_ubu).

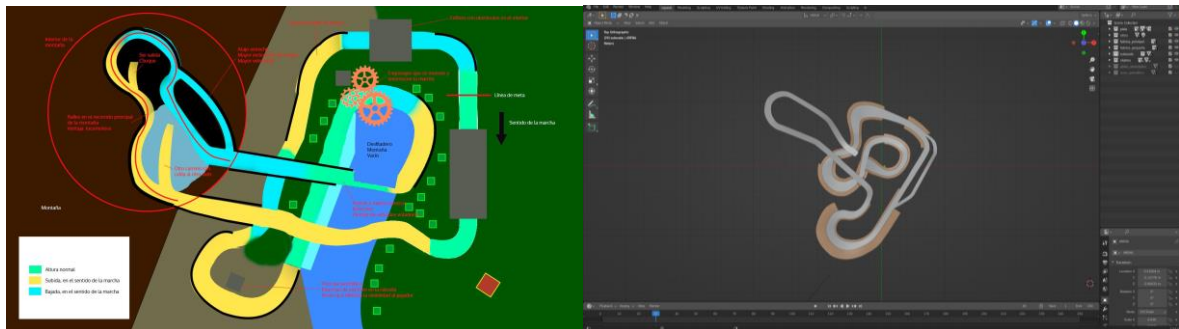


Figura 21. Diseño de elaboración propia (izquierda) y diseño definitivo de la pista (derecha).

Además, se tuvo en cuenta que el diseño se iría adaptando en función del flujo de trabajo y complejidad del mismo en relación a los objetivos y al equipo de trabajo. También se realizó un listado de objetos que se debían modelar y cuáles de ellos iban a tener movilidad y funcionalidad en el escenario. Concretamente, se clasificaron los objetos y máquinas según la cantidad de partes móviles que iban a tener en el escenario. Por último, partiendo de todo lo anterior, se establecieron niveles de calidad (*LOD o level of detail*) para facilitar el proceso posterior de texturizado tras el modelado.

5.2.2. Modelado

Siguiendo la planificación, la fase de modelado es una de las etapas en la que más tiempo se invierte, debido a la carga de trabajo. Por ello, cada uno de los objetos se prefirieron modelar en archivos independientes, con el fin de facilitar el mismo proceso de modelado, el

rendimiento del ordenador y la exportación posterior. También, otra de las ventajas de este método de trabajo es que, a pesar de tener que manejar una mayor cantidad de archivos que ocupan más espacio en el disco, resulta más cómodo modificar los modelos para luego volverlos a importar en el archivo principal. Además, la organización a base de carpetas (o colecciones, como las nombra Blender) facilitó la búsqueda de los objetos.

Para realizar el correcto modelado de los elementos planificados se tuvo en cuenta el estilo que debían de tener. Todos los objetos de la escena tenían que ser de bajo poligonaje, por lo que los acabados excesivamente suavizados y redondeados debían realizarse adecuadamente de manera que el usuario no apreciara si los objetos están suavizados. Por ello, se siguió la “Metodología” aplicando el tipo de modelado poligonal que permite crear formas complejas a través de otras más simples como cubos o cilindros.

En primer lugar, se modeló un pozo petrolífero como prueba de que el poligonaje del modelado era el correcto para la optimización del videojuego y para alcanzar los objetivos. Este se modeló dividiendo la imagen de referencia en varias partes intentando que coincidiesen con las piezas con las que el pozo petrolífero de la realidad está instalado. Cada una de las piezas se creó partiendo de una de las figuras primitivas de Blender, como el cubo o el cilindro, y que luego se modificó teniendo en cuenta que el modelo tenía que ser de bajo poligonaje.



Figura 22. Pozo petrolífero modelado en 3D formado por 368 polígonos.

Seguidamente, se utilizó el primer modelado de la pista para crear los edificios. Partieron de cubos cuyas dimensiones se adaptaron a las que se creían correctas con respecto a la pista ya modelada. Sin embargo, resultaron estar sobredimensionados en relación al resto de elementos que compondrían el escenario en su conjunto, como la montaña y la mina. Por ello, esta primera versión se desechó, dado que se valoró que iba a llevar más tiempo modificar la malla ya existente que crear una nueva. Sin embargo, este modelo sirvió como base para crear las siguientes versiones con la escala corregida y adaptada a las protagonistas, sus vehículos y el resto de los elementos del escenario. Además de esta referencia, el equipo de ÍTACA añadió uno de los personajes a la nueva versión donde se iba a llevar a cabo el modelado y un cubo que fue la representación de la medida de un metro. De esa manera, todos los miembros del equipo desarrollarían su trabajo en base a esa escala establecida y se evitarían correcciones posteriores relacionadas con las dimensiones.

También con la primera versión de la pista del equipo de ÍTACA se adaptó para que las curvas tuvieran cierta inclinación. Aunque se realizaron diversas modificaciones en cuanto al ángulo del peralte de las curvas para aumentarlo o disminuirlo, sirvió de una primera base para la programación. Además, se realizó una primera prueba en Unreal Engine para comprobar si el peralte era el adecuado o no. Después, el equipo de ÍTACA desarrolló una versión más adaptada con el fin de que la programación del videojuego no diera problemas relacionados con el suavizado de la pista y los límites de la misma.



Figura 23. Modelado del escenario y pista peraltada en la primera versión.

El modelado del proyecto en la nueva versión comenzó con la primera distribución de los edificios en el escenario siguiendo el diseño realizado en la etapa anterior de preproducción. Los edificios partieron de un cubo cuyas dimensiones se modificaron teniendo en cuenta tanto el diseño y la pista como las decisiones del equipo de trabajo. Para crear los huecos de las entradas, los edificios se colocaron más abajo del nivel del suelo, de manera que la pista atravesara las estructuras unos milímetros por encima de la cara inferior del cubo. Además, los tejados se crearon como una pieza a parte para que luego la edición de los mismos resultara más sencilla.

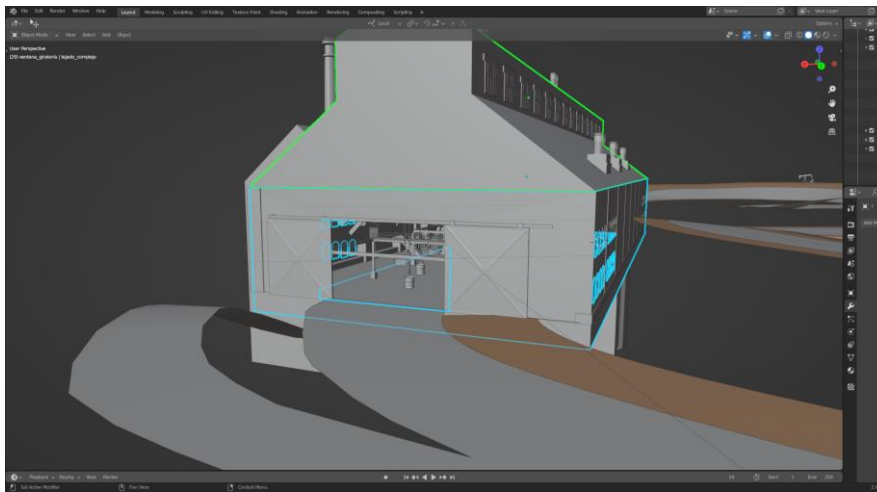


Figura 24. Modelado de la fábrica más grande.

El resto de elementos se modelaron tomando de referencia una imagen. Luego, se creaba una figura primitiva que simularía la parte más grande del objeto. Esa figura, posteriormente, se extruía y se iban editando sus vértices, aristas y caras hasta que tuviera el acabado deseado. Para el resto de las piezas que conformaban el objeto, se fueron generando figuras primitivas independientes de manera que cada una fuera una parte del objeto. Además, esta división de los elementos que componen el objeto tuvo en cuenta las piezas que iban a ser móviles, de manera que estas iban a modelarse independientemente de la base.

Por ejemplo, las ventanas que se ubican en la planta baja de la fábrica principal tienen el cristal central modelado como pieza aparte del marco de la ventana y sus cristales superiores e inferiores. Esto es así para que este cristal del medio bascule tal y como sucede en la realidad con estas ventanas abatibles.

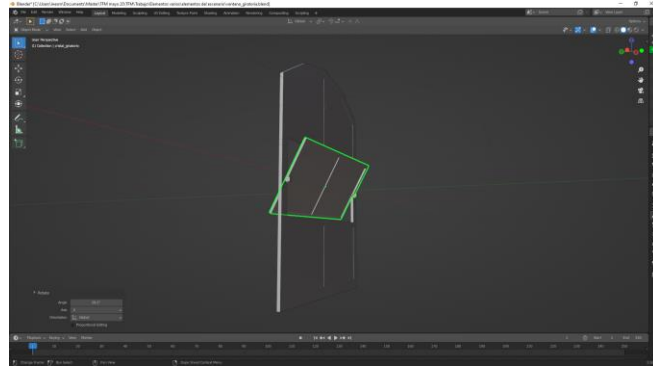


Figura 25. Ventana abatible formada por dos piezas: el marco junto con los cristales superiores e inferiores y el cristal del medio.

Los objetos más simples como barras, chapas y raíles son figuras primitivas a las que únicamente se les modificó la escala para que encajaran con el resto de los elementos del escenario. Las rocas partieron de cubos a los que se le aplicó el modificador *Subdivision surface* con el fin de añadir más vértices que poder modificar su localización para crear rocas con distintas formas y dimensiones. No obstante, para no llegar a un suavizado de la superficie completo solo se aplicó en dos interacciones y que, de ese modo, guardara el modelado de bajo poligonaje. También, para evitar utilizar el modificador de *Subdivision surface* y obtener excesivos polígonos en la malla, se añadieron más vértices según la malla lo requiriera y se suavizaron los mismos en caso de que fuera necesario. Este suavizado se realizó en su justa medida para que no resultaran objetos con la superficie excesivamente lisa.

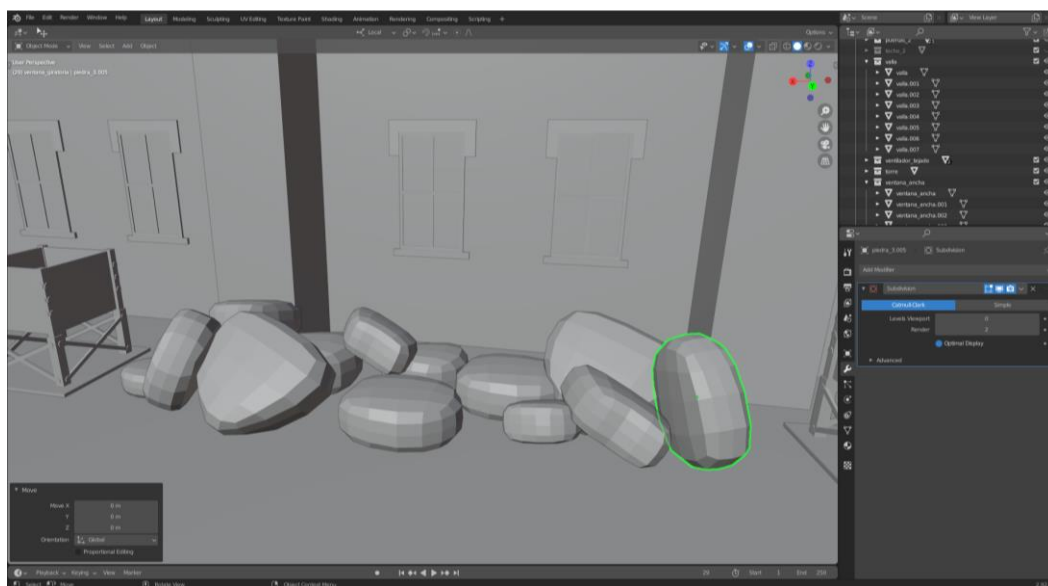


Figura 26. Modelado de las rocas con el modificador *Subdivision surface*.

Las columnas, muros y vallas se modelaron aplicando el modificador *Mirror*. A través de este, se crearon objetos simétricos y se ahorraron número de caras, dado que guardaría la misma estructura que la del lado sobre el que se ha aplicado el modificador. También se aplicó sobre ventanas y puertas. En estos objetos se tuvo en cuenta el punto de origen de los mismos, dado que las ventanas tienen una parte que se puede mover, al igual que las puertas. Por ello, se modificó una vez aplicado el modificador.

Asimismo, las columnas y muros se tuvieron que repetir, por lo que se aplicó el modificador *Array* con el fin de no crear caras excesivas y que quedase simétrico en el menor tiempo de trabajo posible. Este modificador duplicó el elemento en el eje seleccionado cuantas veces fueran necesarias en función de la escala del objeto y el efecto que se quería dar. Por ello, este modificador no se aplicó hasta haber finalizado completamente el modelado, dado que de otra forma, no se hubiera podido modificar si el diseño lo hubiera requerido. Además, este modificador *Array* simplificó el trabajo al modelar las escaleras con sus respectivas barandillas. Únicamente, se modeló un escalón a partir de un plano al que se le dio forma rectangular y se extruyó hacia arriba en el eje Z una de sus aristas más largas. Después se aplicó el modificador *Array* tanto en el eje Z como en el Y y se fue repitiendo para que encajara con la altura del primer piso de la fábrica más grande. Además, se extruyeron las caras para darle profundidad y más realismo. La barandilla que acompaña la escalera partió de un cilindro que se modeló teniendo como referencia el escalón que se modeló previamente. Luego, se aplicó el modificador *Array* tanto en el eje Z como en el Y, al igual que se había realizado con la escalera. Sin embargo, al tener diferentes alturas y ángulos entre el barrote vertical inicial y el que simula un pasamanos, la barandilla no tenía la misma inclinación que la escalera. Por ello, se fueron modificando los parámetros teniendo como referencia la escalera y adaptando los valores de los ejes Y y Z del modificador hasta obtener el resultado deseado.

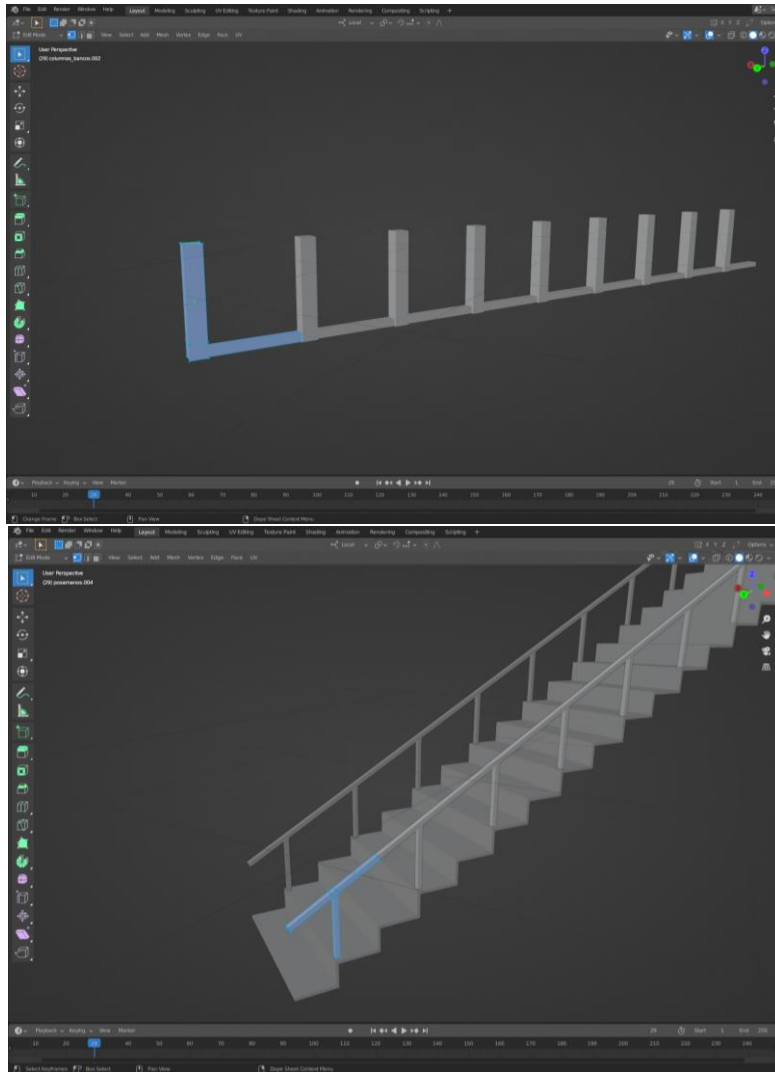


Figura 27. Modelado de las columnas (arriba) y modelado de la escalera y barandilla con el modificador *Array* (abajo).

Para modelar las ruedas dentadas, también se utilizó el modificador *Array*. Primero se modeló un diente de la rueda junto con su hueco al lado suyo y se le aplicó el modificador *Array*. Por otro lado, se creó un círculo de 16 vértices y se redimensionó para que tuviera la escala proporcional al lugar donde se quería colocar. Al elemento en 3D creado previamente, se le aplicó el modificador *Curve*, el cual fusionó el conjunto de dientes al círculo en el eje especificado. Antes de aplicar los modificadores, se ajustaron las interacciones del modificador *Array* y la escala de los dientes. Luego, se aplicaron ambos modificadores. Finalmente, se extruyeron los vértices interiores para rellenar la zona interna de la rueda hasta alcanzar el diámetro deseado.

En el escenario hay otras ruedas dentadas más simples que fueron modeladas sin modificadores. Estas ruedas parten de un círculo de 16 caras, las cuales se extruyeron hacia el exterior del mismo con las medidas que tendrían los dientes de la rueda. Posteriormente, se seleccionaron ocho de esos polígonos intercalándose entre sí y se eliminaron para obtener la forma de rueda dentada. Luego, se rellenó el interior de la misma manera que en el caso anterior: extruyendo los vértices internos hacia dentro. Finalmente, para dar grosor a la malla, se seleccionaron todos los polígonos y se extruyeron hasta obtener la escala deseada.

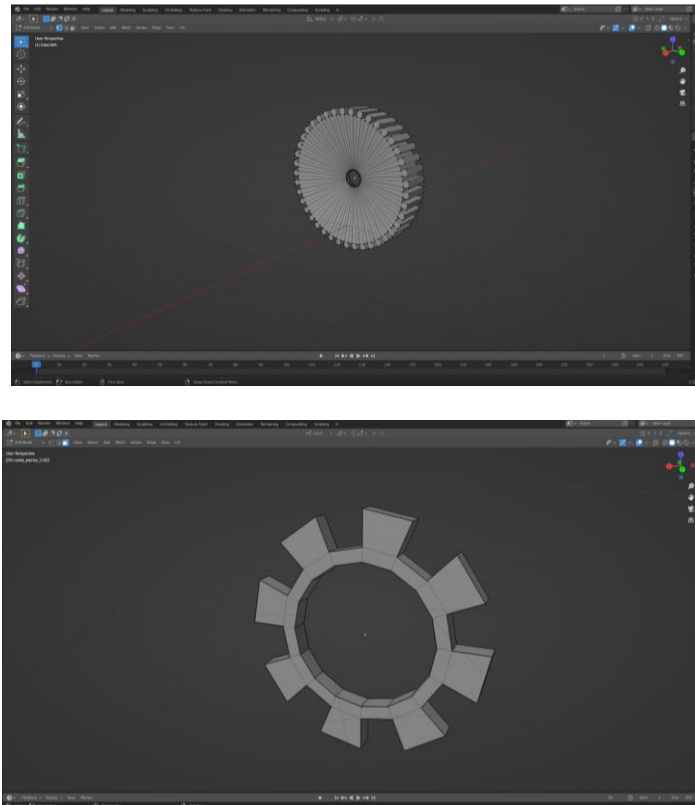


Figura 28. Rueda dentada compleja (arriba) y rueda dentada simple (abajo).

El modelado de las cadenas que se localizan en las grúas de la fábrica más grande combinó los modificadores *Mirror* y *Array*. Partiendo de la figura primitiva de un *torus* de 8 caras, se seleccionaron los vértices de una mitad y se eliminaron. Luego, se aplicó el modificador *Mirror* y se editó la altura del *torus*, de manera que fuera más bajo, tomando forma de elipse horizontal. La mitad reflejada se modificó en cuanto a su posición en el eje X para que actuara como siguiente eslabón. Seguidamente, se aplicó el modificador *Array* y se aumentaron las interacciones para dar una extensión a la cadena más que suficiente. Como esa cadena se iba a duplicar por el número de grúas previstas que se iban a colocar, la longitud de la cadena se

ajustó una vez finalizada la colocación definitiva de las grúas, seleccionando y eliminando los eslabones de la cadena que sobraban.

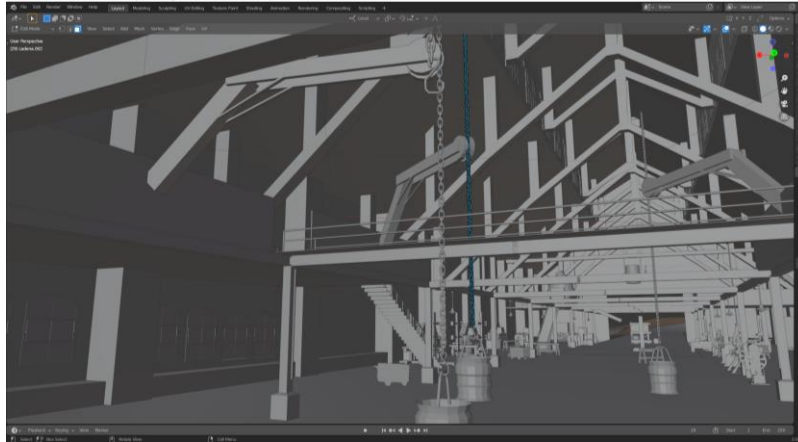


Figura 29. Modelado de la cadena sin haber acortado su longitud.

Para generar detalles como roscas y tuercas se utilizaron figuras primitivas, concretamente cilindros de 6 a 8 caras con el fin de que la malla no tenga un número excesivo de polígonos, dado que aplicando el modificador *Screw* generaría un número excesivo de polígonos para un modelado de bajo poligonaje en el que el detalle se aprecia a través de las texturas.

Los hornos que se encuentran en la planta baja de la fábrica más grande se modelaron por piezas. De manera separada se modelaron la chimenea, las mesas, las puertas y los detalles, para que resultara más sencillo realizar el mapeado y posteriormente aplicar las texturas. Los detalles que se hallan en las puertas se hicieron sobresalir más de lo que en realidad estarían, con el fin de que resaltaran más con respecto al resto de la estructura.

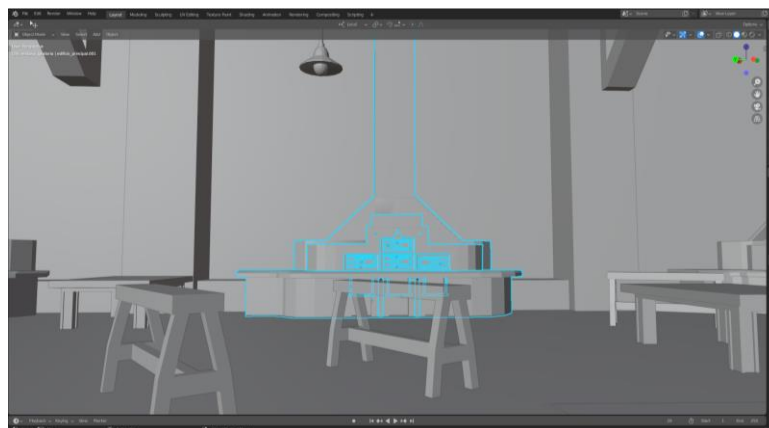


Figura 30. Horno modelado con las piezas independientes.

Con el fin de encajar los elementos sobre otras mallas, se recurrió al modificador *Boolean* en su opción de *difference* para poder generar huecos con la forma del primer objeto en la segunda malla. Este modificador se aplicó en las fachadas de los edificios para generar huecos en las ventanas y que, de esa manera, pudiera modificarse la posición y rotación de la parte móvil de las ventanas. Sin embargo, este modificador ocasionó problemas al no generar los efectos deseados y dar error al aplicarlo sobre la malla. No obstante, otra de las opciones de este modificador, *union*, funcionaba correctamente al aplicarlo sobre la malla. Después, las caras, aristas y vértices del objeto que se añadía a la malla se eliminaban. Este proceso era demasiado lento y había muchos objetos sobre los que aplicar este procedimiento. Por ello, se fueron realizando cortes con la herramienta del cuchillo de Blender siguiendo la forma de las ventanas que previamente se habían duplicado y colocado en la posición en la que se había planificado. Después las ventanas se encajaron en cuanto posición y profundidad respecto a la malla de los edificios y se editaron los huecos de los vértices en el caso que fuera necesario para poder obtener los resultados óptimos. Como este proceso generaba triángulos en la malla que podrían causar problemas a la hora de importar y renderizar el proyecto en Unity se comenzó desde cero la estructura. Las ventanas ya modeladas se añadió un bucle en su marco. Después, se extruyó y se escaló hasta tener un tamaño considerable. Estas nuevas caras se alinearon para que formaran un cuadrilátero y se texturizó. Luego, se duplicaron con el modificador *Array* y se unieron formando una única malla de la estructura sin triángulos.

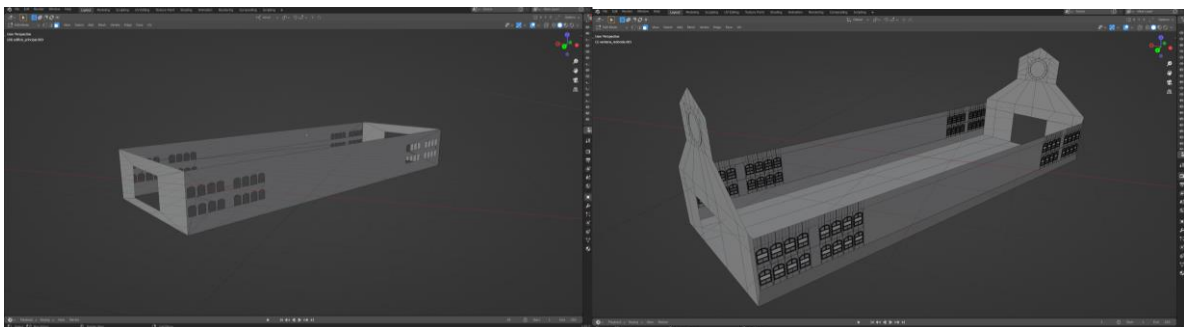


Figura 31. Fachada de la fábrica más grande después de realizar los huecos de las ventanas con el *Boolean* (izquierda) y modelando la fachada partiendo de las ventanas (derecha).

A la mayoría de los objetos y máquinas se les tuvo que aplicar el modificador *Solidify* para atribuirles grosor y un mayor realismo. Sin embargo, en algunos objetos no daba el resultado deseado, por lo que se extruyó la cara y modificando su localización levemente.

La distribución de los elementos en las fábricas se llevó a cabo en la producción, dado que la planificación sirvió de guía para una primera versión. Sin embargo, se fue modificando en

función de las zonas que más visibles iban a resultar para el jugador. De esa manera, la parte de la fachada interna en el circuito tiene más elementos que rellenan su diseño en comparación al lado opuesto. El interior de las fábricas se relleno colocando los objetos de manera que parecieran desordenados, como si estuvieran siendo utilizados, pero siguiendo la lógica de la narración. De ese modo, las máquinas están ordenadas en función del proceso de fabricación de piezas de hierro y obtención del acero. También se tuvo en cuenta que debía haber obstáculos para que el circuito resultara más complejo y entretenido para el jugador. Además, estos objetos tenían que apreciarse con antelación y ser lo suficientemente grandes para que el jugador pudiera sortearlos sin colisionar contra ellos. La puerta de la fábrica más grande se diseñó de esa manera para que se pudiera modificar el tamaño de la entrada y/o salida de la misma en un futuro, dado que, al ser puertas correderas, se puede cambiar la localización para estrechar más el paso inicial e incluso final de la fábrica. Seguidamente a la entrada, se colocaron calderos que el jugador tiene que esquivar. También por su camino tendrá que esquivar otros elementos como estanterías, moldes y contenedores y no podrá acercarse en exceso a los laterales, porque colisionaría contra las máquinas o, en el tramo final, se caería por los huecos. Estas se colocaron en un principio más alejadas de la pista, pero luego se acercaron de modo que el jugador pudiera apreciarlas en mayor medida. Asimismo, se tuvo en cuenta la rotación de estos elementos, según lo establecido en la planificación, para que pudieran apreciarse por el ángulo que era más realista y fueran más reconocibles.

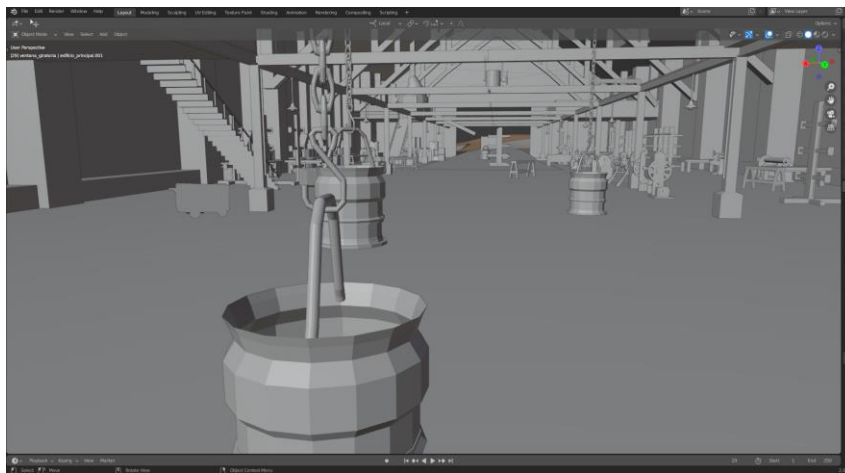


Figura 32. Interior de la fábrica más grande con todos los elementos colocados.

Una vez finalizado el modelado, los elementos se revisaron para eliminar caras, aristas y vértices que no fuesen útiles al objeto modelado. Además, realizar esta revisión reduce los polígonos que texturizar en la siguiente etapa.

5.2.3. Texturizado

La etapa de texturizado fue la más compleja y laboriosa de todas debido a la cantidad de elementos a los que había que aplicar las texturas. Sin embargo, gracias a la planificación de asignación de material a objetos según el que tuvieran en la realidad el texturizado pudo ser más rápido.

Para que los edificios que componen la fábrica simularan las condiciones en las que se encontraría una fábrica siderúrgica en aquella época, se decidió modificar las texturas para que tuvieran manchas simulando el carbón y daños en el material ocasionados por el paso del tiempo y la manipulación de objetos pesados y tóxicos en las fábricas. Para ello, se utilizó el programa Adobe Photoshop CC con su herramienta de pincel o de subexponer y, modificando sus valores de dureza y tamaño, se generaron manchas aleatorias por las texturas que lo requerían, como es el caso de los ladrillos que componen la fachada de los edificios y de las tejas.

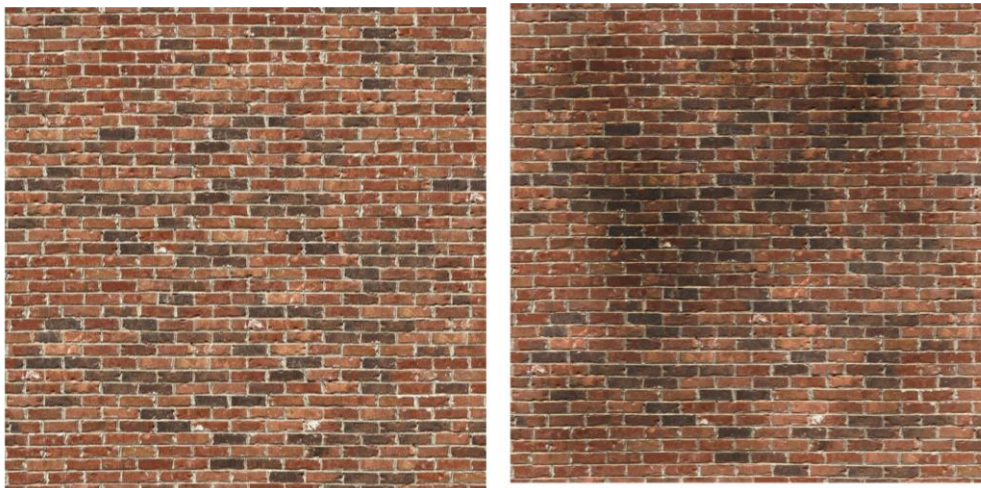


Figura 33. A la izquierda, textura original del ladrillo y, a la derecha, textura del ladrillo con manchas.

Como el modelo sobre el que se iban a aplicar es grande con respecto a las dimensiones de otros objetos, la textura tiene una resolución de 4096x4096 con el fin de que los ladrillos y tejas tuvieran una escala proporcional al de todos los elementos. Con la textura aplicada en Blender, se fue modificando el *UV Mapping* para que generara los resultados más óptimos en cuanto a calidad visual y que se aproximara a la realidad. Para solucionar el problema de la escala de la fachada y reducir el tamaño de los ladrillos, se aumentaron las dimensiones del mapeado UV de las fábricas, de esa manera, los ladrillos parecían tener un menor tamaño, pero conllevaba que la textura de los ladrillos se repitiera un mayor número de veces. Para hacer que no se

notara esta repetición que genera un escaso realismo, se utilizaron objetos ya modelados y las propias manchas de la textura de ladrillos. Interponiendo tanto los objetos como las manchas, la textura creó un efecto de mayor realismo a las fachadas de las fábricas.

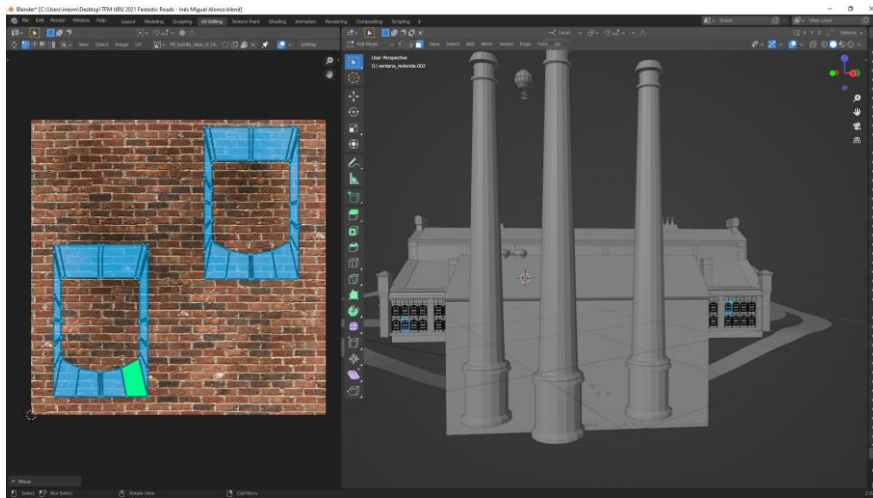


Figura 34. *UV Mapping* agrandado y modificado de dos ventanas de la fachada de la fábrica principal.

Además, para los mapas de textura que no incluía el archivo obtenido de los bancos *online* de texturas, se procesaron en Adobe Photoshop CC a partir del mapa de textura albedo. Para generar el mapa de *roughness* se fueron pintando con el pincel y añadiendo otras formas poligonales la capa en escala de grises en función de mayor o menor rugosidad. Se iluminaron las zonas que mayor rugosidad iban a tener mediante el color blanco o los próximos a él, variando según el detalle de rugosidad del área, y se oscurecieron las zonas que iban a resultar más lisas y que fueran a tener reflexión. También se generó el mapa *normal* mediante la opción que ofrece Adobe Photoshop CC de generar el efecto 3D de mapa de normales y modificando los parámetros para obtener el resultado deseado. El formato de las texturas fue el indicado por el equipo ÍTACA, siendo el JPG todos los mapas de texturas a excepción del *normal* que sería en PNG.

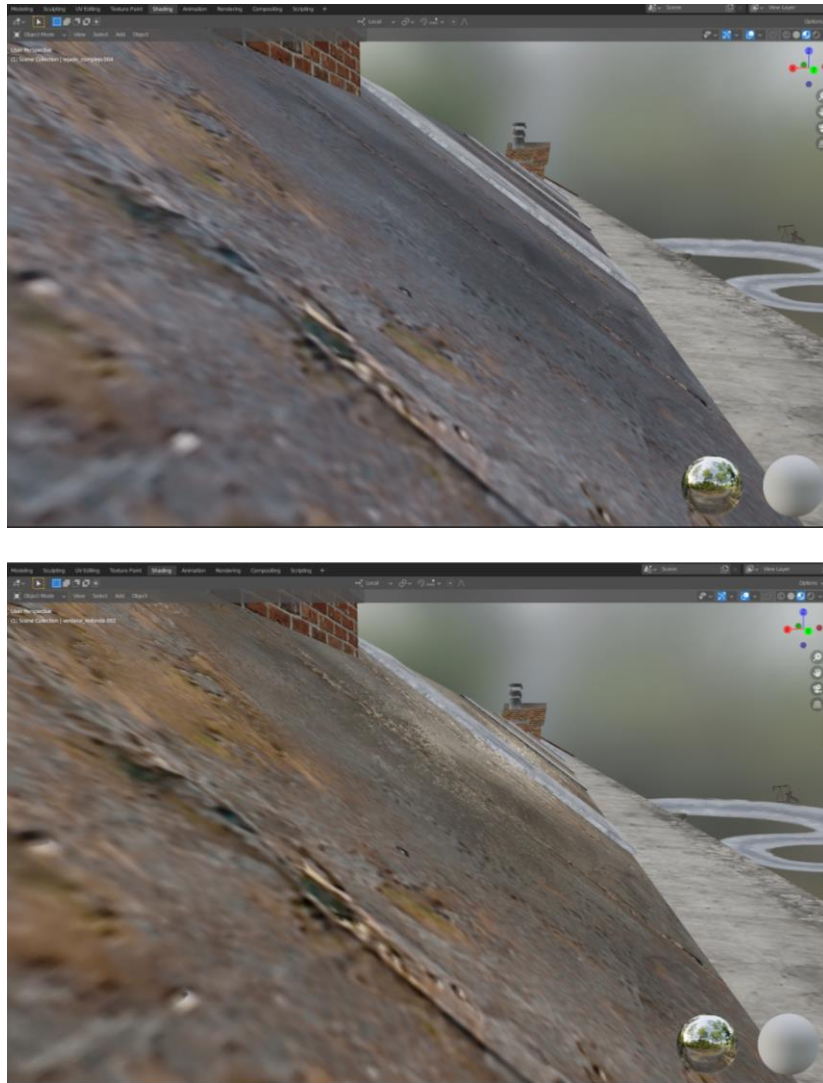


Figura 35. Texturizado final del tejado únicamente con el *base colour* (arriba) y con *roughness* y *normal mapping* aplicados (abajo).

Muchos de los elementos se repiten por lo que lo más conveniente para ahorrar tiempo de trabajo fue aprovechar el mapeado UV del objeto y aplicarle correctamente la textura. Posteriormente, ese objeto se duplicó y se fueron cambiando la posición y rotación del mapeado UV para que la textura no resultara exactamente la misma, sino parecida, aunque teniendo en cuenta que no se tenían que apreciar las costuras y que se debía obtener un resultado coherente y realista.

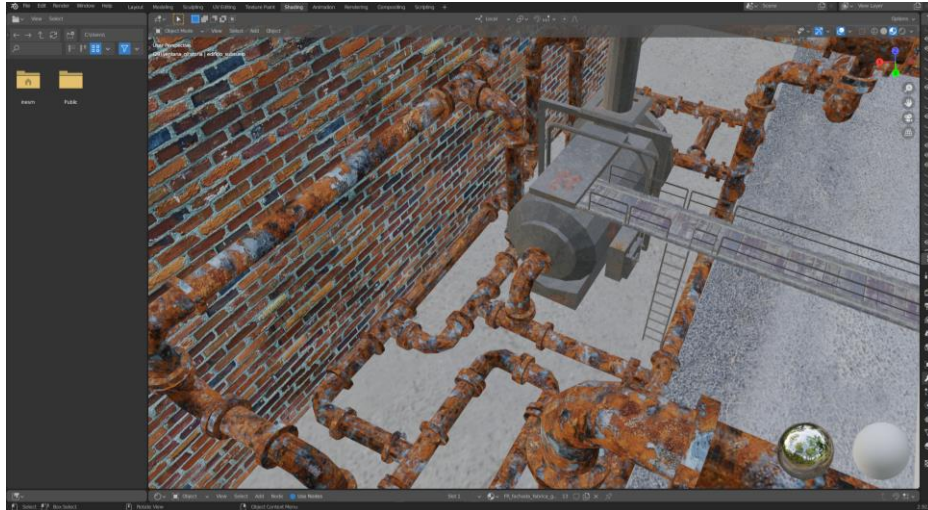


Figura 36. Primer texturizado de la caldera y las tuberías duplicando los objetos y alterando los UV.

Para obtener resultados específicos en algunos de los elementos, se utilizó el otro procedimiento de trabajo de crear la textura en Adobe Photoshop CC a partir del mapeado UV realizado en Blender. Algunos de los objetos sobre los que se aplicó este proceso de trabajo fueron la colada y la burra, dado que se querían obtener resultados diferentes a los del resto de elementos texturizados con el mismo material. Por ello, junto al mapeado UV se añadieron los mapas de texturas albedo y se recortaron en el caso de la burra. Esto fue porque, de esa manera, el mapeado UV en Blender se ajustaría no totalmente a los colores pintados, dejando los bordes de color blanco. Como la burra se trata de una herramienta de la fábrica desgastada, las esquinas de los objetos metálicos quedan más claras en comparación a las superficies que no tienen bordes. Sin embargo, para el resto de los elementos del escenario en los que no se quería conseguir este efecto, las zonas blancas se pintaron en su totalidad con la textura albedo. Esta forma de cubrir totalmente con color la textura es útil no solo para que no se generen líneas blancas indeseadas en la malla, sino también para facilitar el cambio de textura y que no se repita con respecto al resto de los objetos que se han duplicado.

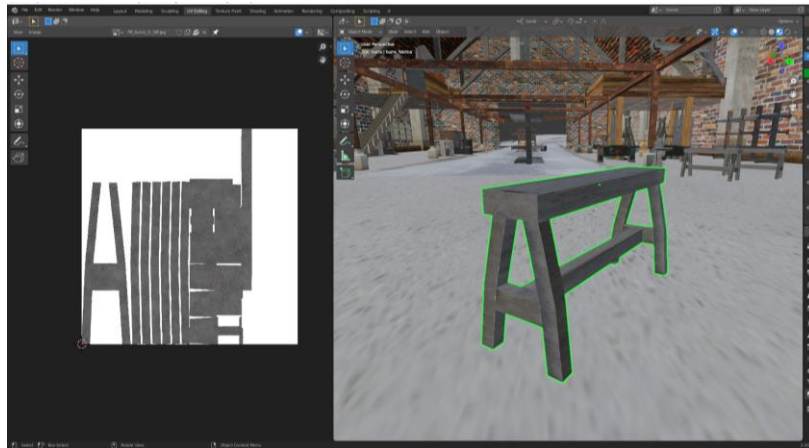


Figura 37. Primer texturizado de la burra.

Para una presentación del modelo, el texturizado de los cristales de las ventanas de las fábricas combinó una textura de metal con manchas y un color grisáceo. Ambos se fusionaron con mezclador de colores (*mix RGB*) en la pestaña de *Shading* de Blender y se aplicaron al *Base colour* del material de los cristales. De esa manera, se aprovecharon las texturas ya importadas y se generó un efecto de cristal que se encontraba sucio y manchado. Además, se aumentó la especularidad para dar realismo y destacar sobre el resto de los objetos metálicos, como es el marco de la ventana. Tanto este elemento como en el caso de otros objetos, hay dos materiales aplicados a la misma malla. Esto se debe a que se le asigna el material a ciertas caras sin cambiar el mapeado UV del objeto. Se tuvo que recurrir a aplicar esta forma de texturizado para objetos en los que resultaba más largo el proceso de separar las piezas. Por ello, se intentó evitar trabajar con una única malla sobre todo en el caso de los objetos sobre los que se iba a aplicar más de una textura.

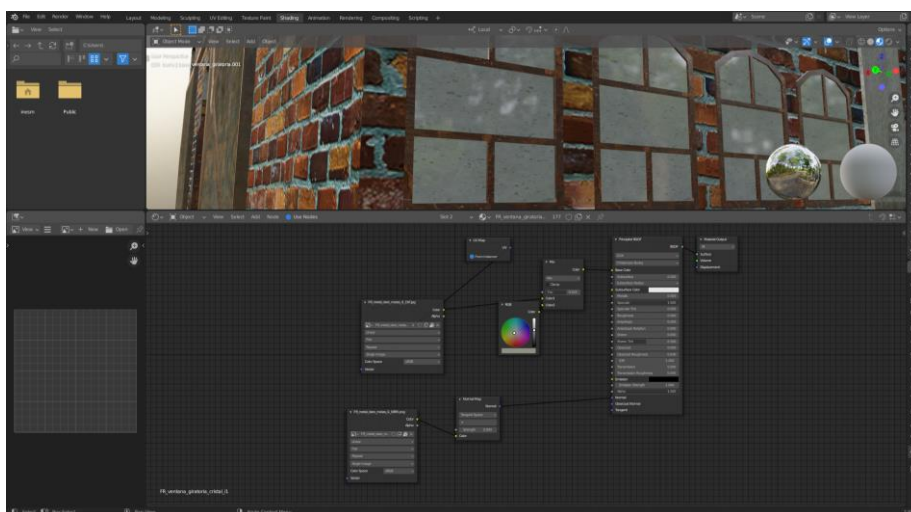


Figura 38. Primer texturizado de las ventanas.

Sin embargo, este texturizado aplicado a los cristales de las ventanas es incorrecto para luego importarlo al proyecto completo en Unity, dado que los nodos no se pueden exportar. Por ello, este texturizado solo es como presentación del modelado de este proyecto.

El resultado final se encuentra en el “Anexo 4”.

6. Conclusiones

Este Trabajo de Fin de Máster ha alcanzado los objetivos propuestos. Se ha logrado diseñar y crear en tres dimensiones parte del escenario que compone el videojuego *Fastastic Roads*, además de optimizar su estructura para que su rendimiento no se vea afectado negativamente. También se ha conseguido trabajar en equipo con una adecuada planificación y distribución de las tareas entre los miembros del ÍTACA. Con este equipo se lograron establecer las bases que unificaron el diseño y dar un sentido a todos elementos a nivel narrativo y técnico.

Concretamente, la etapa de preproducción resultó fundamental para llevar a cabo la siguiente etapa de desarrollo del proyecto. La preproducción no solo consiste en la documentación, sino en seleccionar las técnicas más eficientes para que el resultado fuese el óptimo durante la producción. Además, la preproducción prever los posibles cambios que se iban a realizar, para contar con ese tiempo en el que llevar a cabo las modificaciones. Sin embargo, aunque no tuvo en cuenta los problemas de escala que surgieron, el tiempo de trabajo de modelado en esta parte no se vio afectado en esta parte. Esto es debido a que no se había avanzado excesivamente en el proyecto y el equipo de ÍTACA encontró una solución eficiente para todos los miembros del equipo y que, de ese modo, no se volvieron a tener estos problemas de escala.

La fase de texturizado fue la más larga, debido a su complejidad técnica y a la cantidad de objetos que aplicar las texturas no solo la base, sino otras para generar diversos efectos en función del material en el que está construido en realidad.

A pesar de haber planificado la duración de cada una de las etapas, la fase de producción se alargó, aunque en la preproducción se había previsto que el tiempo de trabajo podía verse aumentado. Esto se debe a que, aunque se habían realizado los diseños, la cantidad de elementos que se iban a modelar podían variar en función de la complejidad del escenario y de las elecciones que se realizaran por parte de todo el equipo de trabajo. Además, se tuvieron que llevar a cabo diversos cambios en cuanto al modelado y a la distribución del escenario para que resultara de bajo poligonaje y que el jugador pudiera tener una experiencia óptima en cuanto a entretenimiento y a nivel visual.

A parte de haber aplicado todo lo aprendido, se ha profundizado en los conocimientos y habilidades del *software* utilizado, concretamente el modelado de bajo poligonaje y sobre todo el texturizado.

En definitiva, el diseño, modelado y texturizado de los elementos que componen la parte del escenario creada cumplen con los requisitos de bajo poligonaje y resultan adecuados a nivel narrativo, estético y de entretenimiento para el jugador. En un futuro se podrá aumentar la complejidad de las animaciones de los objetos modelados, así como incorporar otros elementos e incluso crear otros escenarios los cuales el jugador pueda escoger dentro del videojuego *Fastastic Roads*.

6.1. Análisis del esfuerzo realizado

Este proyecto ha supuesto una gran cantidad de trabajo y esfuerzo, puesto que las horas invertidas en él han sido elevadas en comparación a las horas de trabajo esperadas. Aparte de desarrollar el trabajo de manera autónoma para alcanzar los objetivos propuestos, se ha ido realizando tanto la preproducción como la producción del proyecto de manera paralela con el equipo ÍTACA. Por ello, la comunicación y coordinación con el equipo han resultado indispensables para llevar a cabo el trabajo de manera satisfactoria. Además, con el fin de que el resultado del trabajo fuera el óptimo se han realizado pruebas de diseño y modelado que se han mostrado al equipo para que realizaran sus elecciones y propusieran las modificaciones necesarias en cuestiones de diseño y de funcionalidad del escenario para el videojuego. También se han realizado reuniones de equipo para comunicar los avances y dar opiniones sobre el trabajo de cada uno de los miembros. De esa manera, no solo el trabajo de cada uno estaba al gusto de todos, sino que también se adaptaba a nivel técnico y de funcionalidad al trabajo individual de cada miembro para que todas las partes estuvieran unificadas y los resultados obtenidos fuesen los óptimos.

Este proyecto no solo se ha centrado en el diseño y desarrollo de parte del escenario del videojuego, sino que además se ha adaptado a la situación del equipo para poder optimizar tanto el diseño como la funcionalidad de los elementos que componen el escenario realizado al trabajo del resto de compañeros para que pudieran desarrollarlo satisfactoriamente. Por ello, el diseño se modificó teniendo en cuenta la narrativa, la cual no se vio afectada, pues contaba el mismo proceso. Ejemplo de este caso es modificar los primeros martillos pilones cuya finalidad era aplastar al jugador, pero, al no tener la animación y para no perder la funcionalidad ni la interacción con el jugador en esa zona, se reemplazaron por unos martillos pendulares que empujan al jugador contra los laterales de la fábrica. Además, se tuvo que corregir la escala de los elementos del escenario en la primera versión del proyecto, debido a que los edificios

estaban sobredimensionados en relación a los otros elementos del escenario y a las protagonistas y sus vehículos. Aunque se tuvo que realizar un modelado desde cero en una nueva versión, este primer trabajo sirvió de base, dado que ya se tenían técnicas asentadas que fueron más eficientes a la hora de construir los edificios y corregir algunos errores que facilitarían el texturizado en la fase posterior.

A pesar de las modificaciones en cuanto a diseño y modelado, la etapa que más tiempo ha llevado dentro de la producción ha sido el texturizado, debido a la carencia de conocimientos en ese ámbito y a la cantidad de objetos que se debían texturizar buscando la manera de equilibrar rendimiento y calidad visual. Sin embargo, gracias a ello, se han aumentado los conocimientos y habilidades en cuanto a texturizado a través de imágenes se refiere.

En los siguientes gráficos se muestra el tiempo invertido en horas en cada fase y concretamente en la etapa de producción, a la que más tiempo se dedicó.

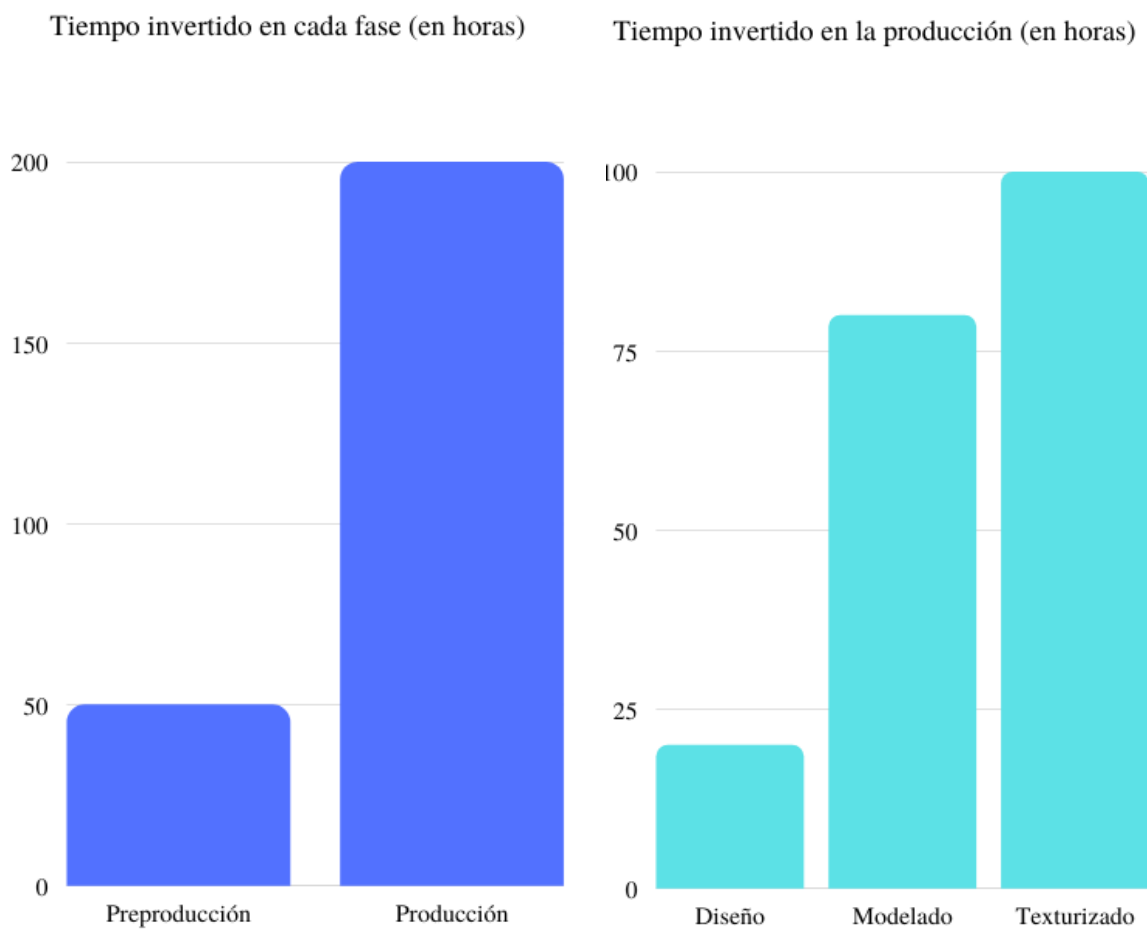


Figura 39. Gráficos del tiempo invertido en cada fase y en producción en horas.

7. Resumen

Este Trabajo Fin de Máster consiste en el diseño y creación de parte del escenario del videojuego de carreras *Fastastic Roads* desarrollado por el grupo ÍTACA de la Universidad de Burgos. Este videojuego pretende dar visibilidad a cuatro mujeres pioneras en distintas áreas de la ingeniería desde mediados del siglo XIX hasta la primera mitad del siglo XX. El escenario objeto de este proyecto se ambienta en esa época para modelar en tres dimensiones dos fábricas en las que se lleva a cabo un proceso de extracción, procesado de hierro, fabricación de piezas metálicas y, además, obtención del acero. De forma paralela, dos Trabajos Fin de Grado junto con otros miembros de ÍTACA desarrollan la programación, imagen del videojuego y el resto del modelado de los elementos que integran el escenario.

El diseño del escenario del videojuego requiere de una amplia preproducción, para que el desarrollo dure el menor tiempo posible. Esta fase de producción se lleva a cabo en Blender, *software* en el que se realiza el modelado y texturizado de los objetos. El modelado es de bajo poligonaje con formas simples y escaso detalle que permite que el escenario no afecte de manera negativa el rendimiento del videojuego. También los objetos y composición del escenario se han adaptado para facilitar la programación de las primeras versiones, pudiéndose utilizar en un futuro como elementos que tengan una interacción más directa con el jugador.

Durante todo el proceso de trabajo se pondrán en práctica los conocimientos aprendidos en el Máster en Comunicación y Desarrollo Multimedia para que el diseño y el modelado de los elementos estén optimizados para su implantación en el videojuego.

Palabras clave: *videojuego de carreras; modelado 3D; escenario 3D; diseño de videojuego; fábrica 3D.*

8. Abstract

This Master's Final Project consists of the design and creation of part of the scenario for the racing video game *Fastastic Roads* developed by the ÍTACA group of the University of Burgos. This video game aims to give visibility to four pioneering women in different areas of engineering from the mid-19th century to the first half of the 20th century. The scenario object of this project is set at that time to model two factories in three dimensions in which a process of extraction, iron processing, manufacture of metal parts and, also, obtaining steel is carried out. In parallel, two Final Degree Projects together with other members of ÍTACA develop the programming, the image of the video game and the rest of the modeling of the elements that make up the scenario.

The design of the video game scenario requires extensive pre-production, so that the development lasts as little time as possible. This production phase is carried out in Blender, software in which the modeling and texturing of the objects is carried out. The modeling is low poly with simple shapes and little detail that allows the setting not to negatively affect the performance of the video game. Also the objects and composition of the scenario have been adapted to facilitate the programming of the first versions, being able to be used in the future as elements that have a more direct interaction with the player.

Throughout the work process, the knowledge learned in the Master in Communication and Multimedia Development will be put into practice so that the design and modeling of the elements are optimized for their implementation in the video game.

Keywords: *racing video game; 3D modeling; 3D scenario; video game design; 3D factory.*

9. Bibliografía y fuentes

3D Ace. (23 de noviembre de 2017). 3D modeling techniques in games. *3d-ace*. <https://3d-ace.com/press-room/articles/3d-modeling-techniques-games>

Aldabaldetrecu, P. (1 de febrero de 2002). Evolución técnica de la máquina-herramienta. Reseña histórica. *Interempresas*. <https://www.interempresas.net/MetalMecanica/Articulos/1435-Evolucion-tecnica-de-la-maquina-herramienta-Resena-historica.html>

Aldric. (7 de abril de 2020). The Evolution of Racing Games (Part 1). *Antidote*. <https://antidote.gg/evolution-of-racing-games/>

Apperley, T. H. (2006). Genre and game studies: Toward a critical approach to video game genres. *Simulation & Gaming*, 37(1), 6–23. doi: 10.1177/1046878105282278

Arbonés, Á. (10 de julio de 2018). Del 2D al 3D: cómo el videojuego pasó de la imaginación a solo valorar los números. *Caninomag*. <https://www.caninomag.es/del-2d-al-3d-o-como-el-videojuego-paso-de-la-imaginacion-a-solo-valorar-los-numeros/>

Arteneo (10 de abril de 2017). ¿Conoces Xbrush 3D? ¿Para qué se utiliza? *Arteneo*. <https://www.arteneo.com/blog/httpswww-arteneo-com-estudiar-curso-zbrush-art-3d-escultura-digital/>

Bernabeu, A. (2007). *Estrategias de diseño estructural en la arquitectura contemporánea. El trabajo de Cecil Balmond* [Tesis doctoral, Departamento de Estructuras de Edificación, Escuela Técnica Superior de Arquitectura, Universidad Politécnica de Madrid]. http://oa.upm.es/910/1/Alejandro_Bernabeu_Larena.pdf

Berzosa, C. (2008). Los modelos de desarrollo tardíos. *Revista de Economía Mundial*, 18, 51-63. <https://www.redalyc.org/pdf/866/86601805.pdf>

Blender (2021). *Manual*. <https://docs.blender.org/manual/es/dev/index.html>

Boullón, A. (2009). Evolución tridimensional en la representación visual de los videojuegos y su repercusión en la jugabilidad. *Revista Comunicación*, 1, 116-133. http://www.revistacomunicacion.org/pdf/n7/articulos/a8_Evolucion_tridimensional_en_la_presentacion_visual_de_los_videojuegos_y_su_repercusion_en_la_jugabilidad.pdf

Bueno, D. (2018). *El diseño y arte en los videojuegos* [Trabajo de Fin de Grado, Escuela universitaria de informática, Grado en Ingeniería de Software, Universidad Politécnica de Madrid]. http://oa.upm.es/50803/1/TFG_DARIO_BUENO_GUTIERREZ.pdf

Cameron, R. (1985). A New View of European Industrialization. *The Economic History Review*, 38(1), new series, 1-23. doi:10.2307/2596641

Candil, D. (26 de marzo de 2011). Juegos de coche que hicieron historia. Parte 2. *Motorpasion*. <https://www.motorpasion.com/cine-tv-juegos/juegos-de-coches-que-hicieron-historia-parte-2>

Cobbett, R. (23 de septiembre de 2017). From shareware superstars to the Steam gold rush: How indie games conquered the PC. *PC Gamer*. <https://www.pcgamer.com/from-shareware-superstars-to-the-steam-gold-rush-how-indie-conquered-the-pc/>

Concept Car Credit (s. f.). Car racing video games: the ultimate history. *Concept Car Credit*. <https://www.conceptcarcredit.co.uk/car-racing-video-games-the-ultimate-history/>

Culture of gaming (27 de noviembre de 2017). The evolution of indie games and why they're great. *Culture of gaming*. <https://cultureofgaming.com/the-evolution-of-indie-games-and-why-theyre-great/>

Dunlop, R. (2014). *Production pipeline. Fundamentals for film and games*. Routledge.

Educba. (s.f.). Design – Key differences tutorial. *Educba tutorials*. <https://www.educba.com/design/design-tutorials/design-key-differences-tutorial/>

Esteves, A. (1 de septiembre de 2003). Evolución de los procesos de corte y deformación de la chapa. *Interempresas*. <https://www.interempresas.net/Deformacion-y-chapa/Articulos/8647-Evolucion-de-los-procesos-de-corte-y-deformacion-de-chapa.html>

Flores, S. (2016). *Métodos numéricos para el diseño de curvas y superficies en ingeniería naval* [Trabajo Fin de Grado, Escuela Técnica Superior de Ingeniería Naval y Oceánica, Universidad Politécnica de Cartagena]. <http://hdl.handle.net/10317/5706>

García, J. (3 de octubre de 2015). *Mega Drive vs. Super Nintendo. La batalla de los 16 bits*. IGN España. <https://es.ign.com/super-nintendo/96766/feature/mega-drive-vs-super-nintendo>

García, J. A. (2015). *La evolución gráfica de los videojuegos. Desde la aparición de consolas de sobremesa* [Trabajo, Universidad Rey Juan Carlos]. https://dibujourjc.files.wordpress.com/2015/03/juan-antonio-garcia-cac3b1as_6061852_assignsubmission_file_la-evolucic3b3n-grc3a1fica-de-los-videojuegos.pdf

Gilmore, C., Moraes, H. (2002). *Gerencia de la calidad en salud*. Gerencia en Salud. Escuela nacional de salud pública. Ministerio de salud pública.

Gómez, A. (2016). *Desarrollo de un videojuego: Flame knights chronicles*. [Trabajo Fin de Grado, Escuela Politécnica Superior, Universidad de Jaén]. http://tauja.ujaen.es/bitstream/10953.1/3365/1/TFG_G%C3%B3mez_L%C3%B3pez%2CAlejandro.pdf

González, N. (2018). Motores de videojuegos. *Universidad Católica “Nuestra Señora de la Asunción”*. *JezuAzarru.com*. <http://jeuazarru.com/wp-content/uploads/2018/11/Motores-de-Videojuegos.pdf>

Grubler, A. (1995). *Industrialization as a Historical Phenomenon*. *IIASA Working Paper*. IIASA, Laxenburg, Austria: WP-95-029. <https://core.ac.uk/download/pdf/33895537.pdf>

Herraez, J.V. (12 de enero de 2011). Historia de los videojuegos: un viaje por la segunda mitad del siglo XX. *Cultura informática, entretenimiento, consolas y videojuegos*. <https://histinf.blogs.upv.es/2011/01/12/historia-de-los-videojuegos-un-viaje-por-la-segunda-mitad-del-siglo-xx/>

IFP. (13 de julio de 2020). ¿Qué hace un modelador 3D? *Innovación en Formación Profesional, IFP*. <https://www.ifp.es/blog/que-hace-un-modelador-3d>

Kaur, A. (2003). *Texture mapping using tiled textures* [Tesis. Visualization Sciences, Office of Graduate Studies of Texas A&M University]. <https://core.ac.uk/download/pdf/147122936.pdf>

Kent, S.L. (2016). *La gran historia de los videojuegos*. B DE BOOKS.

Lobo, Á. (2000). *Baking en Blender* [Trabajo de Fin de Carrera, Departamento de Teoría de la señal y comunicaciones, ETSIS Telecomunicación]. http://oa.upm.es/48491/1/PFC_ANGEL_LOBO_ARANGUEZ.pdf

Llurba, Ll. (11 de octubre de 2017). Del pixelado a la alta definición: La evolución de los gráficos en los videojuegos. *Alphr, Tecnología con un toque diferente*.
<http://alphr.es/cultura/la-sorprendente-evolucion-de-los-graficos-en-los-videojuegos/>

Mac Donald, R. (28 de enero de 1998). Resident Evil 2 Review for PlayStation. *GameSpot*.
<https://www.gamespot.com/reviews/resident-evil-2-review/1900-2549080/>

Manrubia Pereira, A. M. (2014). El proceso productivo del videojuego: fases de producción. *Historia y Comunicación Social*, 19, 791-805.
https://doi.org/10.5209/rev_HICS.2014.v19.45178

Martín Angelina, C. y Ballejo, D. (2012). Desarrollo de videojuegos 1: Arquitectura de motor de videojuegos. *Universidad de Castilla-La Mancha*.

Martínez, A. (20 de abril de 2017). Low poly. El arte de lo minimalista. *Invergestudios*.
<https://www.invergestudios.com/blog/el-arte-low-poly/>

Martínez, G. (1997). Acero. Capítulo III. La producción del acero. *La ciencia para todos, Fondo de Cultura Económica*, 2.
<http://bibliotecadigital.ilce.edu.mx/sites/ciencia/volumen2/ciencia3/080/htm/acero.htm>

McDermott, W. (2018). *The PBR guide: a handbook for physically based rendering*. Allegorithmic.

Melero, S. (20 de julio de 2017). Gráficos y perspectiva en los videojuegos. *MeriStation, As*.
https://as.com/meristation/2017/07/19/reportajes/1500444000_167709.html

MeriStation (3 de marzo de 2012). Papyrus; la historia de un mito. *MeriStation, As*.
https://as.com/meristation/2006/01/02/reportajes/1136161980_036375.html

Morales, G. A., Nava, C., Fernández, L.F., Rey, M. A. (2010). *Procesos de desarrollo para videojuegos*, 7 (36-37), 25-39. <https://dialnet.unirioja.es/servlet/articulo?codigo=3238114>

Moscovici, T. (2014). *KAZ: Pushing the Virtual Divide* [Documental]. Sony Computer Entertainment (SCE); Polyphony Digital; Industry Films; Tamir Moscovici.

Nestorlink. (3 de junio de 2020). La evolución gráfica en los videojuegos Parte II. *Area xBox*. <https://areabox.com/index.php/2020/06/03/la-evolucion-grafica-en-los-videojuegos-parte-ii/>

Oleva, S. (5 de noviembre de 2017). Pole Position. *Ultimate History of video games!* <https://ultimatehistoryvideogames.jimdofree.com/pole-position/>

Paíno Plaza, A. (2004). *Estudio técnico-económico de una acería de horno eléctrico de arco con el uso de chatarra como materia prima* [Trabajo Final de Carrera, Escola Universitària d'Enginyeria Tècnica Industrial de Barcelona, UPC]. <http://hdl.handle.net/2099.1/5193>

Pajot, L., Swirsky, J. (2011). *Indie Game: The Movie* [Documental]. BlinkWorks Media.

Palmieri, J. (2019). Evolution of Nascar game. The 64-bit era. *Jon Palmieri, Web Art & Design*. https://acadweb.hvcc.edu/~j-palmieri3/webart_fall/final_site/64-bit.html

Pecheux, H. (1910) *Précis de metallurgie*. J.B. Baillièere et Fils, Paris.

Pluralsight. (8 de abril de 2014). What's the difference? A comparison of modeling for games and modeling for movies. *Pluralsight*. <https://www.pluralsight.com/blog/film-games/whats-the-difference-a-comparison-of-modeling-for-games-and-modeling-for-movies>

Reina, J. (10 de septiembre de 2018). Breve historia del 3D y los videojuegos. *V-Art, Escuela Audiovisual*. <https://v-art.es/breve-historia-del-3d-y-los-videojuegos/>

Reina, J. (4 de junio de 2018). ¿Qué es Zbrush y para qué sirve? Una breve introducción. *V-Art, Escuela Audiovisual*. <https://v-art.es/que-es-zbrush-y-para-que-sirve-una-breve-introduccion/>

Ros, I. (2 de junio de 2019). Pasado, presente y futuro de los videojuegos a través de ocho generaciones de consolas. *Muy Computer*. <https://www.muycomputer.com/2019/06/02/historia-videojuegos-consolas/>

Rubi 3D. (24 de enero de 2014). 70. Triángulos y quads. *Rubi3D*. <https://rubi3d.com/70-triangulos-y-quads/>

Sacchi, A. (3 de febrero de 2015). Creación de modelos 3D para Gaming, ¿Empezamos? *Slashmobility*. <https://slashmobility.com/blog/2015/02/creacion-de-modelos-3d-para-gaming-empezamos/>

Salen, K., Zimmerman, E. (2003). *Rules of play: game design fundamentals*. MIT Press.

Schvab, L. (2011). *Guía didáctica. Capítulos 1, 2, 3 y 4. Máquinas y herramientas*. Buenos Aires, Argentina: Editorial Encuentro <http://www.inet.edu.ar/wp-content/uploads/2012/11/maquinas-y-herramientas.pdf>

Shmuplations (2001). Kasco and the electro-mechanical golden age conducted and published in 2001 by CVS Odyssey. *Shmuplations*. <http://shmuplations.com/kasco/>

Silverman, D. (5 de marzo de 2013). Manual básico para desarrolladores de juegos: una descripción general del modelado 3D en juegos. *Game Development*.

<https://gamedevelopment.tutsplus.com/es/articles/3d-primer-for-game-developers-an-overview-of-3d-modeling-in-games--gamedev-5704>

Solo, M. (21 de octubre de 2018). La primera era del 3D: La generación que lo cambió todo. *MeriStarion*, As. https://as.com/meristation/2018/10/19/reportajes/1539948499_015444.html

Stearns, P. N. (2018). *The Industrial Revolution in World History* (4th ed.). Routledge.

Tokio School. (8 de enero de 2020). Todos los elementos del modelado para videojuegos. *Tokio School*. <https://www.tokioschool.com/noticias/elementos-modelado-videojuegos/>

Trazos (2021). Retopología en 3D. ¿Qué es y para qué sirve? *Trazos*. <https://trazos.net/retopologia-en-3d-que-es-y-para-que-sirve/>

Unity Documentation (2016). Manual de Unity. Guía de buenas prácticas para assets de arte. *Unity 3D*. <https://docs.unity3d.com/es/530/Manual/HOWTO-ArtAssetBestPracticeGuide.html>

Unity Technologies (2016). Manual de Unity. Optimizando el rendimiento gráfico. *Unity 3D*. <https://docs.unity3d.com/es/530/Manual/OptimizingGraphicsPerformance.html>

Vallejo, D., Martín, C. (2013). *Desarrollo de videojuegos: Arquitectura del motor de videojuegos* (2ª ed.). LibroVirtual.org.

Vélez, I. (10 de junio de 2014). La fórmula de Geoff Crammond. *Deux ex machina, Video games magazine*. <http://deusexmachina.es/la-formula-de-geoff-crammond/>

Videojuegos Horacio (s. f.). Juegos Commodore 64: Buggy Boy. *Videojuegos Horacio*.
<https://www.videojuegoshoracio.com/juegos-c64-buggy-boy/>

Ward, A. (20 de diciembre de 2019). “Winning Run” Namco (1988) History of Arcade Racing.
Part 18. *Three Fields Entertainment*.
<https://www.threefieldsentertainment.com/2019/12/20/winning-run-namco-1989-history-of-arcade-racing-part-18/>

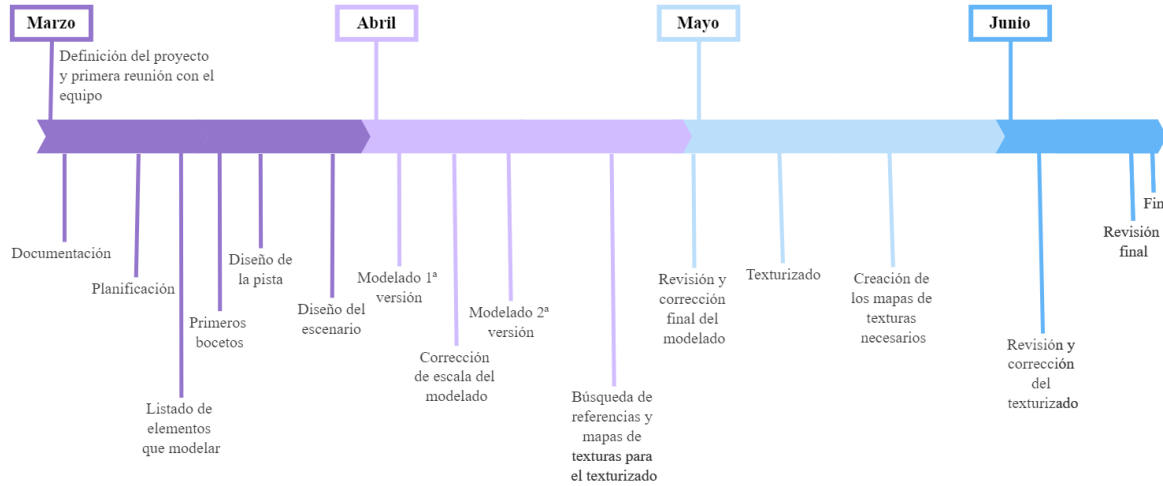
10. Índice de imágenes

Figura 1. Mini Drive (1959). Fuente: Shumplations.	10
Figura 2. Turbo (1981). Fuente: Sega Retro.	11
Figura 3. Pole Position (1982). Fuente: Time Graphics.	11
Figura 4. Out Run (1986). Fuente: Mod DB.	12
Figura 5. Winning Run (1988). Fuente: Xtreme Retro.	13
Figura 6. Motor Toon Grand Prix (1994). Fuente: Zonared.	14
Figura 7. Gran Turismo (1998). Fuente: Gran-turismo.	14
Figura 8. Triangulación de la malla de esfera con topología correcta. Fuente: Vendrell, E. (2018).	16
Figura 9. Cuatro levels of detail de un objeto según la distancia con respecto al jugador. Fuente: 3djuegos.	17
Figura 10. Street Fighter II: The world warrior en Super Nintendo (1992) y PlayStation (1998). Fuente: VCDECIDE.	19
Figura 11. Pantalla de trabajo en Blender y mallas primitivas.	20
Figura 12. Curva Bézier (arriba) y Nurbs (abajo) en Blender.	21
Figura 13. Panel de modificadores de una malla en Blender.	22
Figura 14. Shading en Blender con una textura bitmap aplicada.	23
Figura 15. PBR y mapas de textura para un objeto metálico. Fuente: Substance Academy. .	24
Figura 16. Mario Kart 8. Fuente: Guías Nintendo.	34
Figura 17. Byoshock Infinite. Fuente: HobbyConsolas.	34
Figura 18. Grabados de una mortajadora y una fresadora del libro “Précis de métallurgie” de Pecheux (1910).	35
Figura 19. Ejemplos de mapas de textura procesados en Deep Dream Generator de una columna.	36
Figura 20. Boceto del escenario por ÍTACA. Fuente: Instagram de ÍTACA (@itaca_ubu). .	38

Figura 21. Diseño de elaboración propia (izquierda) y diseño definitivo de la pista (derecha).	38
Figura 22. Pozo petrolífero modelado en 3D formado por 368 polígonos.	39
Figura 23. Modelado del escenario y pista peraltada en la primera versión.....	40
Figura 24. Modelado de la fábrica más grande.....	41
Figura 25. Ventana abatible formada por dos piezas: el marco junto con los cristales superiores e inferiores y el cristal del medio.....	42
Figura 26. Modelado de las rocas con el modificador Subdivision surface.	42
Figura 27. Modelado de las columnas (arriba) y modelado de la escalera y barandilla con el modificador Array (abajo).	44
Figura 28. Rueda dentada compleja (arriba) y rueda dentada simple (abajo).	45
Figura 29. Modelado de la cadena sin haber acertado su longitud.....	46
Figura 30. Horno modelado con las piezas independientes.....	46
Figura 31. Fachada de la fábrica más grande después de realizar los huecos de las ventanas con el Boolean (izquierda) y modelando la fachada partiendo de las ventanas (derecha).	47
Figura 32. Interior de la fábrica más grande con todos los elementos colocados.....	48
Figura 33. A la izquierda, textura original del ladrillo y, a la derecha, textura del ladrillo con manchas.....	49
Figura 34. UV Mapping agrandado y modificado de dos ventanas de la fachada de la fábrica principal.	50
Figura 35. Texturizado final del tejado únicamente con el base colour (arriba) y con roughness y normal mapping aplicados (abajo).....	51
Figura 36. Primer texturizado de la caldera y las tuberías duplicando los objetos y alterando los UV.	52
Figura 37. Primer texturizado de la burra.	53
Figura 38. Primer texturizado de las ventanas.	53
Figura 39. Gráficos del tiempo invertido en cada fase y en producción en horas.	57

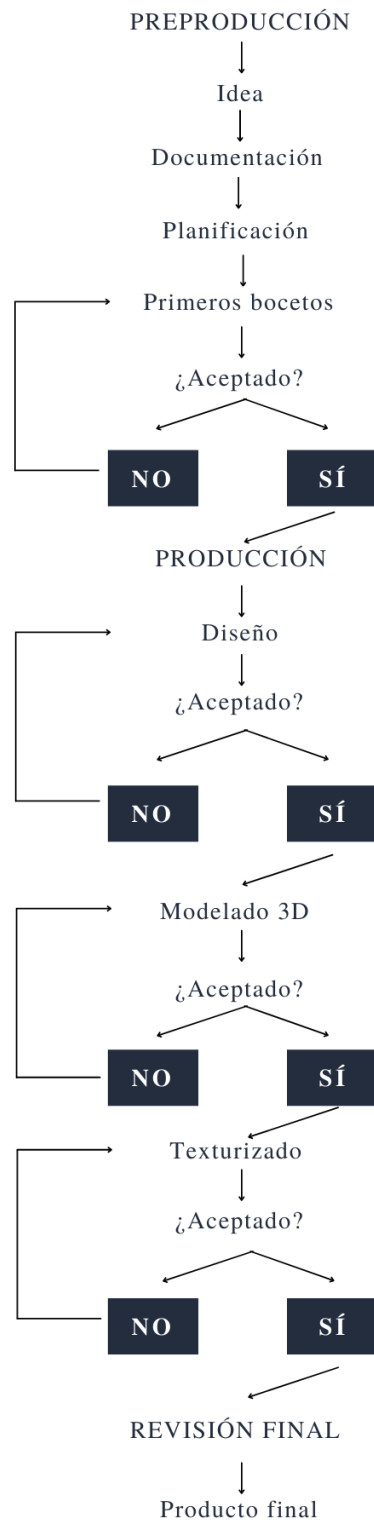
11. Anexos

Anexo 1. Esquema de las etapas y planificación

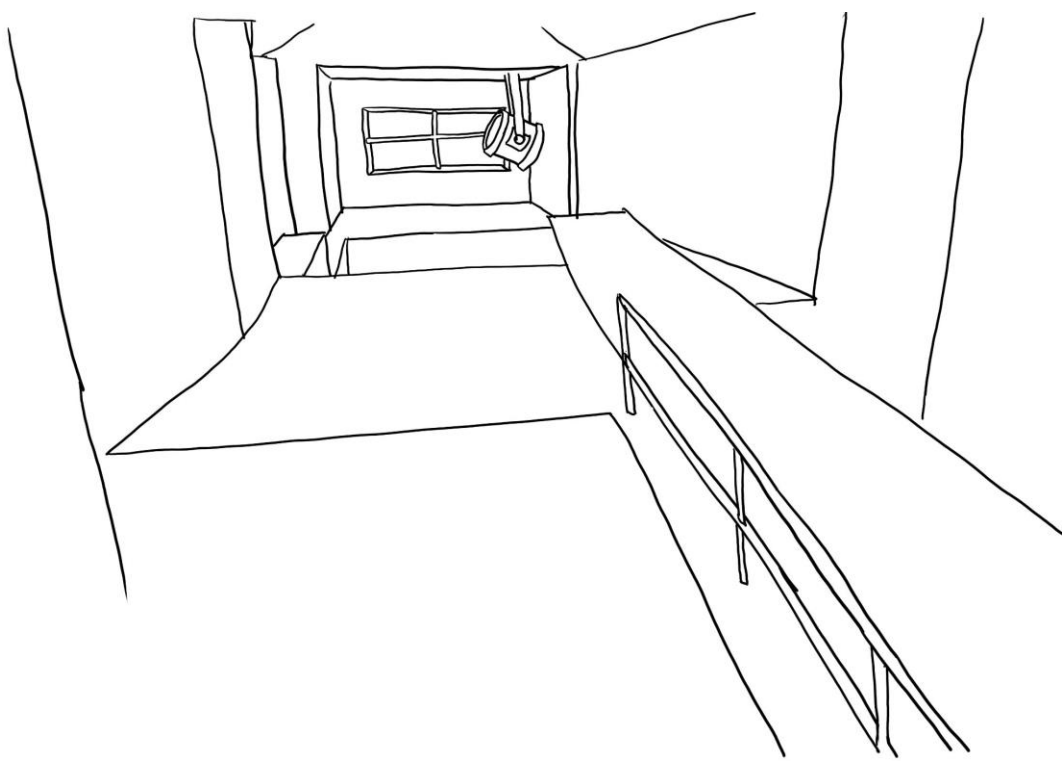
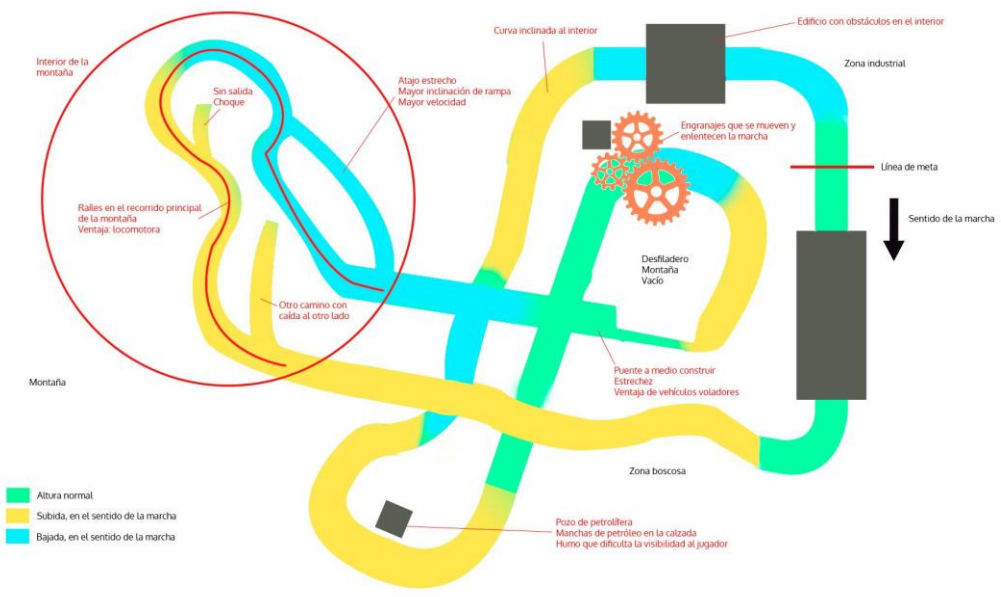


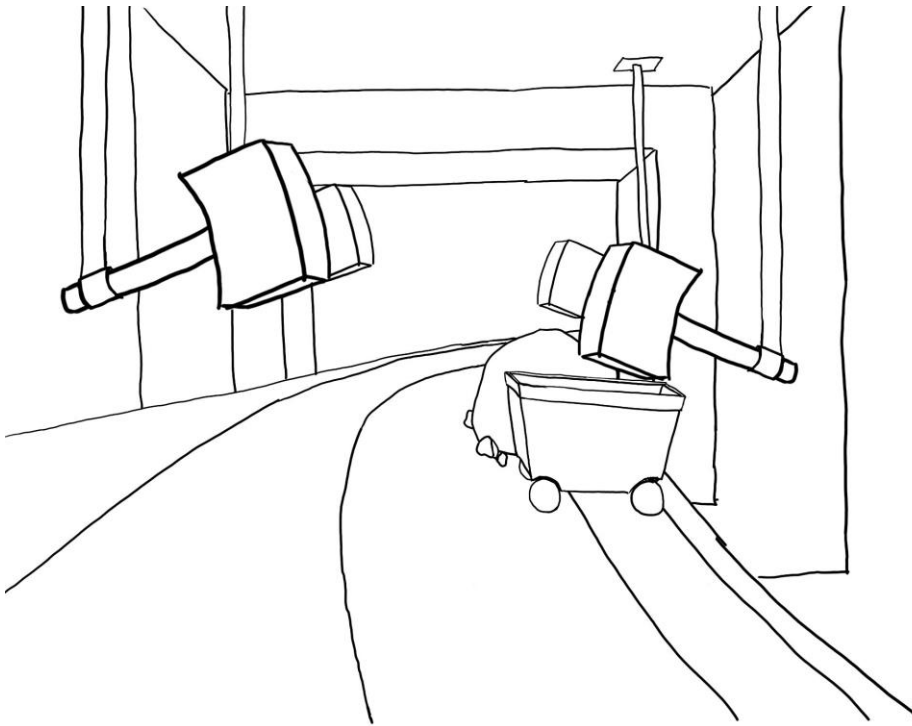
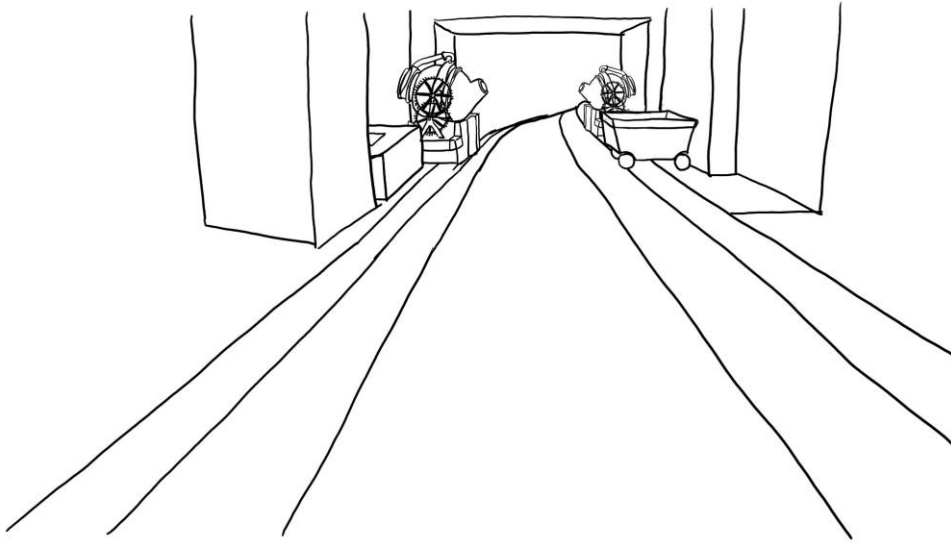
	Marzo	Abril	Mayo	Junio
Idea	●			
Documentación	●			
Planificación	●			
Diseño	●			
Modelado 3D		●		
Texturizado			●	
Revisión final				●
Fin				●

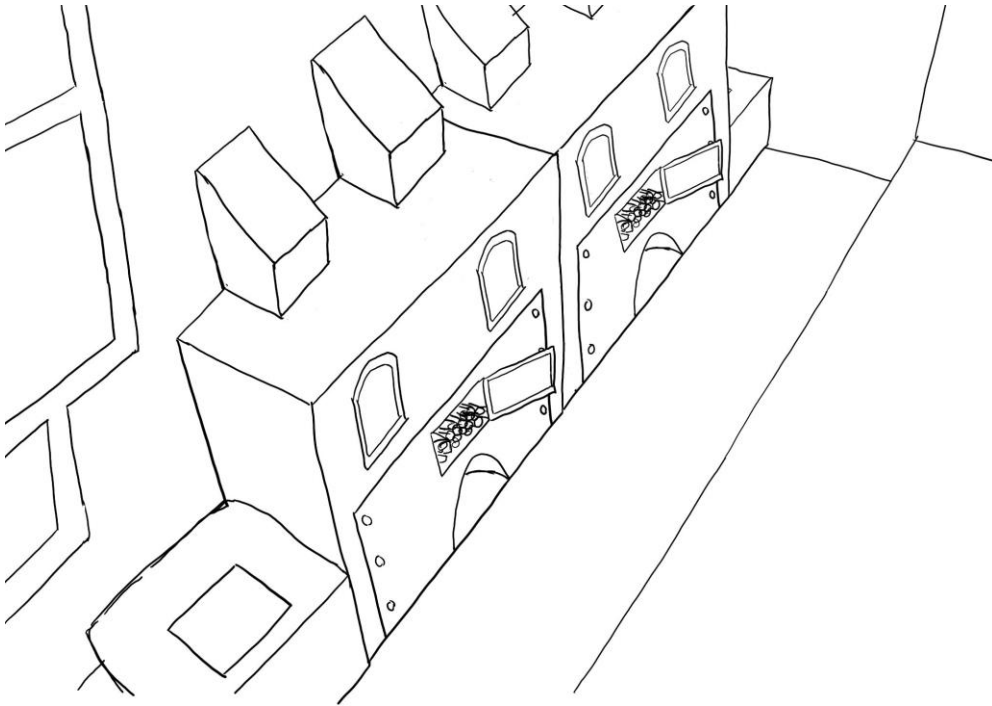
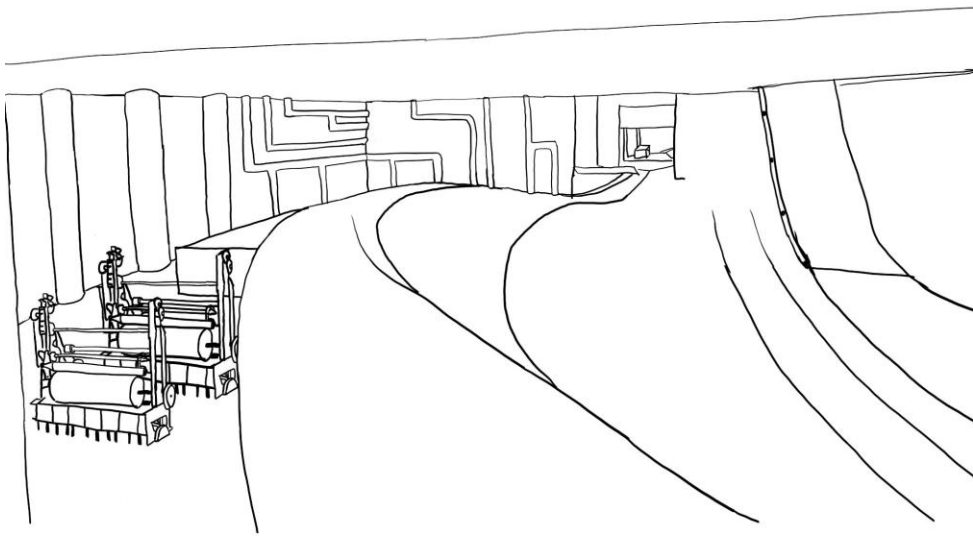
Anexo 2. Flujograma

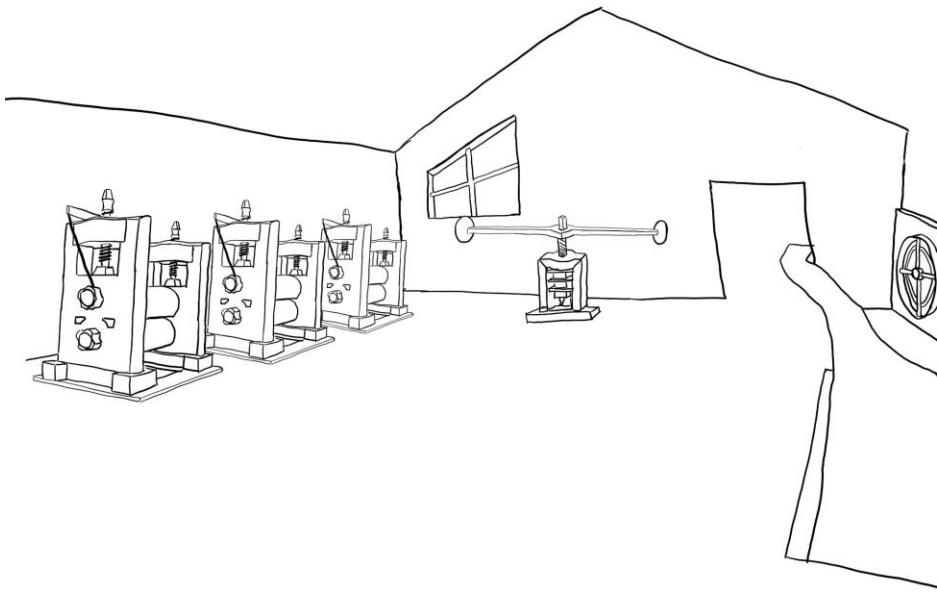


Anexo 3. Bocetos del escenario









Anexo 4. Material multimedia