

Optimización y mejora de funcionalidades de gemelos digitales en Unity3D mediante renderizado de Alta Definición.

Grado en Ingeniería Electrónica, Robótica y Mecatrónica

Mención en Robótica y Automatización

Realizado por:

Jaime Sebastián Billón Rodríguez

Bajo la tutorización de:

Tutor: Prof. Ricardo Vázquez Martín

Cotutor: Louis Fröhlich



UNIVERSIDAD
DE MÁLAGA



Departamento de Ingeniería de Sistemas y Automática.

ESCUELA DE INGENIERÍAS INDUSTRIALES

UNIVERSIDAD DE MÁLAGA

Málaga

Enero 2024

Resumen

El presente trabajo de final de grado se centra en la mejora de las características de renderizado y funciones de la visualización de gemelos digitales en Unity, una plataforma de desarrollo para videojuegos y aplicaciones interactivas. El proyecto se desarrolló en el departamento de Automatización de EDAG Production Solutions GmbH & Co. KG, parte del grupo EDAG Engineering Group AG, y se basó en el EDAG Automation Studio (EDAS), una herramienta de visualización en 3D y flujo de materiales para plantas de producción.

El objetivo principal es abordar los desafíos actuales y futuros relacionados con la simulación y visualización en entornos virtuales, en particular en el ámbito de los gemelos digitales. Estos gemelos digitales son utilizados en diversas disciplinas, como medicina, ingeniería, arquitectura y entretenimiento, y su mejora tiene un potencial significativo para impactar positivamente en estos campos. El enfoque es permitir una visualización y simulación más efectiva y precisa, y optimizar los procesos sin interrumpir el funcionamiento del componente real, evitando pérdidas económicas.

El trabajo también se centra en la importancia de Unity como entorno versátil para la creación de gemelos digitales, utilizados en una amplia gama de aplicaciones. La mejora en las características de renderizado y las funciones de los gemelos digitales en Unity es crucial para satisfacer las expectativas crecientes de los usuarios y mantener la competitividad en el mercado. Además, la investigación busca aprovechar las nuevas capacidades tecnológicas para ofrecer simulaciones más precisas, interactivas y detalladas.

En el trabajo se destaca también la función de resaltado como una característica importante en el desarrollo de la herramienta EDAS en Unity. Esta función permite a los usuarios identificar y enfocar áreas específicas dentro del modelo virtual, mejorando la interactividad y la eficiencia en el análisis y la toma de decisiones. Esta característica es esencial para una visualización detallada y una comprensión más profunda de los modelos, y juega un papel importante en la optimización de los procesos de diseño y revisión.

Palabras clave

Unity3D, gemelo digital, renderizado, renderizado de Alta Definición, renderizado Estándar, shaders, técnica de resaltado, HDRP, Built-in, Outline, Unity industrial, Puesta en marcha Virtual.

Abstract

This final degree project focuses on enhancing the rendering capabilities and functionalities of digital twins in Unity, a development platform for video games and interactive applications. The project was conducted at the Automation department of EDAG Production Solutions GmbH & Co. KG, a part of the EDAG Engineering Group AG, and was based on the EDAG Automation Studio, a 3D visualization and material flow tool for production plants.

The main objective is to address current and future challenges related to simulation and visualization in virtual environments, particularly in the realm of digital twins. These digital twins are utilized in various disciplines such as medicine, engineering, architecture, and entertainment, and their improvement holds significant potential to positively impact these fields. The focus is on enabling more effective and accurate visualization and simulation, optimizing processes without disrupting the operation of the real component and avoiding economic losses.

The work also emphasizes the importance of Unity as a versatile environment for creating digital twins used in a wide range of applications. Enhancing rendering features and functionalities of digital twins in Unity is crucial to meet the growing expectations of users and to maintain competitiveness in the market. Additionally, the research seeks to leverage new technological capabilities to provide more precise, interactive, and detailed simulations.

The paper also emphasises the highlighting function as an important feature in the development of the EDAS tool in Unity. This feature allows users to identify and focus on specific areas within the virtual model, enhancing interactivity and efficiency in analysis and decision-making. This feature is essential for detailed visualization and deeper understanding of models, playing a significant role in optimizing design and review processes.

Keywords

Unity3D, Digital Twin, rendering, High Definition Render Pipeline, Built-in Render Pipeline, shaders, Highlight function, HDRP, Outline, Industrial Unity, Virtual Commissioning.

Siglas y acrónimos

A

API: Application Programming Interface. Interfaz de programación de aplicaciones.

B

BIM: Building Information Modeling. Modelado de información de construcción.

C

CPU: Central Processing Unit. Unidad central de procesamiento.

D

DTA: Digital Twin Aggregate. Agregado de Gemelo Digital.

DTI: Digital Twin Instance. Instancia de Gemelo Digital.

DTP: Digital Twin Prototype. Prototipo de Gemelo Digital.

E

EDAG: EDAG Engineering Group AG.

EDAG PS: EDAG Production Solutions GmbH & Co. KG.

EDAS: EDAG Automation Studio.

F

FIBC: Flexible Intermediate Bulk Containers. Contenedores a granel intermedios flexibles.

FXAA: Fast Approximate Anti-Aliasing. Aproximación Rápida de Antialiasing.

G

GPU: Graphics Processing Unit. Unidad de procesamiento de gráficos.

H

HD: High Definition. Alta Definición.

HDR: High Dynamic Range

HDRP: High Definition Render Pipeline. Renderizado de Alta Definición.

HLSL: High-Level Shader Language. Lenguaje de sombreado de alto nivel.

HVAC: Heating, Ventilation and Air Conditioning. Calefacción, ventilación y aire acondicionado.

I

I+D. Innovación y Desarrollo.

IA. Inteligencia Artificial.

IG. Iluminación Global.

IIoT: Industrial Internet of the Things. Internet de las cosas industrial.

IoT: Internet of the Things. Internet de las cosas.

L

LDR: Low Dynamic Range

LPPV: Light Probe Proxy Volume. Volumen proxy de sonda luminosa.

LWRP: Lightweight Render Pipeline. Renderizado ligero.

M

MSAA: Multisample anti-aliasing. Antialiasing multimuestra.

P

PBR: Physics-Based Rendering. Renderizado basado en la física.

PDP: Physical to Digital to Physical. Físico a digital a físico.

PLC: Programmable Logic Controller. Controlador lógico programable.

R

RA. Realidad Aumentada.

S

SH. Spherical Harmonics. Esféricos armónicos.

SRP: Scriptables Render Pipelines. Renderizado a través de scripts.

SSAO: Screen Space Ambient Occlusion. Oclusión ambiental del espacio de pantalla

SSR: Screen Space Reflection. Reflexión del espacio de la pantalla.

T

TFG: Trabajo Fin de Grado.

U

UI: User Interface.

URP: Universal Render Pipeline. Renderizado universal.

V

VFX: Visual Effects. Efectos visuales.

Índice general

Contacto	I
Resumen	II
Palabras clave	III
Abstract	IV
Keywords	V
Siglas y acrónimos	VI
1 Introducción	1
1.1 Introducción	1
1.2 La empresa EDAG	1
1.2.1 EDAG Engineering Group AG	1
1.2.2 EDAG Production Solutions GmbH & Co. KG	2
1.2.3 Departamento de automatización de EDAG PS	3
1.2.4 EDAG Automation Studio	4
1.3 Motivación del trabajo	5
1.4 Objetivos	6
1.5 Estructura del documento	7
2 Estado del arte: Gemelos digitales en la Industria 4.0	8
2.1 Industria 4.0	8
2.1.1 Características de la Industria 4.0	9
2.1.2 Tecnologías de la Industria 4.0	10
2.1.3 Objetivos de la Industria 4.0	12
2.1.4 Soluciones gracias a la Industria 4.0	13
2.2 Gemelos digitales	15
2.2.1 Introducción a los gemelos digitales	15
2.2.2 Características de los gemelos digitales	16
2.2.3 Tipos de gemelos digitales	17
2.2.4 Ventajas que aportan los gemelos digitales	19
2.2.5 Aplicaciones de los gemelos digitales	21
2.3 Unity 3D	22
2.3.1 Servicios de Unity	23
2.3.2 Renderizadores de malla en Unity	25

2.3.3	Sombreadores en Unity	26
2.3.4	Postprocesamiento	30
3	Renderizados de Unity 3D	33
3.1	Introducción al renderizado	33
3.2	Renderizado Estándar o Built-in	33
3.2.1	Introducción al renderizado Estándar	33
3.2.2	Técnicas de renderizado en el renderizado Estándar	35
3.2.3	Orden de renderizado en el renderizado Estándar	36
3.3	Renderizado Universal	37
3.3.1	Introducción al renderizado Universal	37
3.3.2	Características del renderizado Universal	38
3.3.3	Ventajas de la Implementación del renderizado Universal	40
3.4	Renderizado de Alta Definición	41
3.4.1	Introducción al renderizado de Alta Definición	41
3.4.2	Características propias del renderizado en Alta Definición	41
3.4.3	Beneficios del renderizado de Alta Definición	47
3.4.4	Técnicas de renderizado en Alta Definición	48
3.5	Comparativa entre los renderizados Universal y Alta Definición	49
3.5.1	Análisis de características técnicas y diferencias	49
3.5.2	Rendimiento y fidelidad visual	51
3.5.3	Complejidad y facilidad de uso	51
3.5.4	Elección del renderizado para el presente trabajo	51
4	Entorno del gemelo digital	53
4.1	Características de la visualización con el renderizado Estándar	53
4.1.1	Componentes del gemelo digital	53
4.1.2	Limitaciones en la visualización del gemelo digital	60
4.2	Conversión de la visualización del gemelo digital a Alta Definición	61
4.2.1	Pasos realizados para aplicar la conversión	61
4.3	Comparación de la visualización en ambos renderizados	63
5	Funcionalidades de la visualización del gemelo digital en Alta Definición	76
5.1	Análisis de problemas con renderizado de Alta Definición	76
5.1.1	Diferencias técnicas entre materiales y sombreadores en los renderizados Estándar y Alta Definición	77
5.1.2	Función de resaltado	78
5.2	Propuestas para solucionar los problemas	79
5.2.1	Solución a través del uso de activos de Unity	79
5.2.2	Solución a través de los sombreadores de Unity	81
5.2.3	Solución mediante la reprogramación de un activo	84
5.3	Validación	88
6	Conclusiones y Líneas de trabajo futuras	90
6.1	Conclusiones	90
6.2	Líneas de trabajo futuras	91

6.2.1	Mejora de la función de resaltado	91
6.2.2	Resaltado mediante el uso de sombreadores	92
6.2.3	Uso de otros activos de Unity	92
6.2.4	Manual de uso	92

Capítulo 1

Introducción

1.1. Introducción

EDAG Production Solutions GmbH & Co. KG es una empresa perteneciente al grupo EDAG Engineering Group AG con sede en Fulda, Hesse, Alemania. Dentro de esta sede y, concretamente, dentro del departamento de Automatización, en el equipo de Puesta en Marcha Virtual, ha sido desarrollado el presente TFG.

1.2. La empresa EDAG

1.2.1. EDAG Engineering Group AG

EDAG Engineering Group AG (de ahora en adelante, EDAG) es una sociedad anónima alemana fundada en 1969 por Horst Eckard en Groß-Zimmern, un municipio perteneciente al distrito de Darmstadt-Dieburg. Esta empresa es considerada como uno de los socios de ingeniería independiente más grande del mundo dentro de la industria automotriz, siendo reconocida durante todo el proceso de desarrollo: desde el desarrollo estratégico, iniciado en la fase conceptual, hasta la planificación y optimización de la producción [1].

Actualmente cuentan con más de ocho mil empleados en sesenta emplazamientos repartidos en 19 países alrededor del mundo. Los beneficios de la compañía ascendieron a setecientos dos millones de euros el pasado año 2023. [2, 3]

El grupo EDAG se divide en las siguientes empresas:

- **EDAG Engineering.** Se encarga del desarrollo completo de vehículos, realización de conceptos de diseño, desarrollo de módulos y funciones de vehículos, validación completa de vehículos mediante la realización de pruebas, gestión de datos de procesos y de productos, gestión y mantenimiento de la calidad, ingeniería de pruebas, pruebas y construcción de vehículos y oferta de soluciones postventa. [2, 4]
- **EDAG Electronics.** Realiza apoyo a nivel mundial para clientes con experiencia en el desarrollo del sector eléctrico y electrónico, incluye desarrollo de conceptos y soluciones innovadoras para futuras tecnologías de vehículos, como conducción altamente automatizada, conectividad, movilidad eléctrica, digitalización, redes y servicios móviles. [2].
- **EDAG Productions Solutions.** Es un proveedor de servicios de ingeniería del grupo EDAG e independiente de los fabricantes de plantas y sistemas. Ejecuta grandes proyectos en todo el mundo de acuerdo con los requisitos del cliente, asumiendo también la responsabilidad como planificador o contratista general. [2]
- **Feynsinn.** Se trata de consultoría para la mejora de procesos de desarrollo digital. Ayuda a aumentar la eficiencia en el procesamiento de datos relevantes para el desarrollo y a optimizar las interfaces entre desarrollo, producción y marketing. Además, proporciona ideas, métodos y herramientas que tienen como objetivo incrementar la motivación y la eficiencia del equipo de desarrollo. [2]
- **Trive.me.** Trabaja la innovación en productos, servicios, soluciones individuales y estrategias, lanzamiento y operación de servicios, plataformas, sitios web y aplicaciones propias y garantía de un rápido progreso del proyecto. [5]
- **EDAG Aeromotive.** Crea soluciones innovadoras y sostenibles, así como ayuda en la producción, validación y desarrollo de entornos de sistemas para aplicaciones de pruebas, análisis y evaluación en la aeronáutica. [6]
- **EDAG Akademie.** Ofrece una amplia gama de cursos y seminarios especializados para clientes del sector automovilístico, así como ayuda en la preparación del material de seminarios y documentación. [2]

1.2.2. EDAG Production Solutions GmbH & Co. KG

EDAG Production Solutions GmbH & Co. KG (de ahora en adelante, EDAG PS), empresa perteneciente al grupo EDAG y con sede central en Fulda, Alemania, se encarga de la gestión de proyectos, el análisis de factibilidad, la planificación de la producción, la ingeniería y el diseño de sistemas, la implementación de instalaciones y la tecnología de montaje, la simulación de flujo de materiales, la automatización

de instalaciones, la optimización de la producción, los servicios de ingeniería de seguridad y la garantía de calidad. [7]

EDAG PS cuenta con más de mil trescientos empleados en veintitrés emplazamientos repartidos alrededor del mundo y los beneficios de la compañía ascendieron a ciento ocho millones de euros el pasado año 2023. [3, 8]

1.2.3. Departamento de automatización de EDAG PS

Dentro de la empresa EDAG PS hay numerosos departamentos, entre los que se encuentra el departamento de Automatización. En la Figura 1.1 se pueden observar los distintos servicios ofrecidos por el departamento de Automatización de EDAG PS. Estos contienen todo el proceso de producción, compuesto por el diseño del soporte físico (hardware), el diseño de los programas (software), el control de calidad y la puesta en marcha virtual y finalizando con la puesta en marcha en planta.

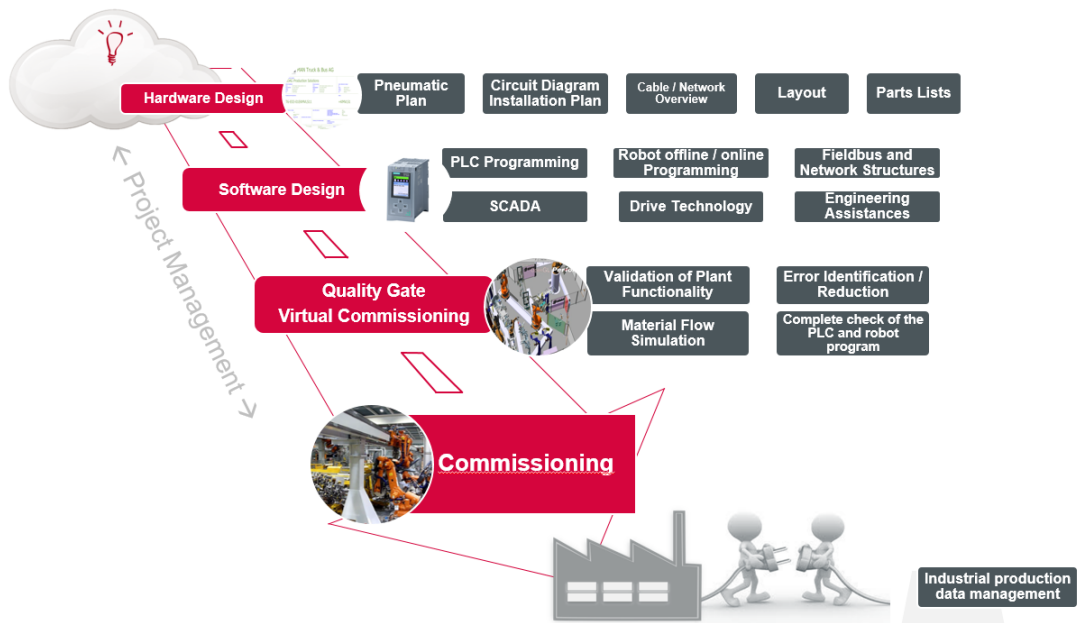


Figura 1.1: Servicios del departamento de automatización de EDAG PS. Fuente: Intranet de EDAG PS.

El presente proyecto se ha desarrollado dentro del departamento de Automatización, dentro de la sección de Puesta en Marcha Virtual (Virtual Commissioning).

1.2.4. EDAG Automation Studio

El EDAG Automation Studio, conocido brevemente como EDAS, es un desarrollo interno de la empresa EDAG PS. Se utiliza principalmente en la puesta en marcha virtual para proyectar la visualización en 3D y el flujo de materiales de una planta de producción o de un proceso de producción.

La Figura 1.2 muestra como se conforma el entorno EDAS. Se trata de una aplicación ejecutable que permite mostrar una visualización 3D de los gemelos digitales que se hayan cargado dentro de él. Esta aplicación permite navegar a través del árbol de contenido, seleccionar los objetos disponibles y moverse dentro del entorno visual generado. También permite realizar conexiones con otros programas, conectar con el entorno de robot y con señales provenientes del PLC y otras herramientas de simulación complementarias como WinMod.



Figura 1.2: Visualización del entorno EDAS. Fuente: EDAG PS.

1.3. Motivación del trabajo

La realización del presente trabajo está fundamentada en la necesidad de abordar los desafíos actuales y futuros que rodean la simulación y visualización en entornos virtuales, específicamente en el ámbito de los gemelos digitales.

Los gemelos digitales son ampliamente utilizados en diversas disciplinas, desde la medicina y la ingeniería hasta la arquitectura y el entretenimiento. La mejora de las características de renderizado y funciones de un gemelo digital tiene un potencial significativo para impactar positivamente en estos campos, permitiendo una visualización y simulación más efectiva y precisa. Además, permite el tratamiento y la optimización de los procesos sin necesidad de utilizar el componente real e interrumpir su funcionamiento, lo que evita pérdidas económicas significativas.

Unity, como una de las principales plataformas de desarrollo para videojuegos y aplicaciones interactivas y cada vez más presente en el ámbito de la ingeniería, ha demostrado ser un entorno versátil para la creación de gemelos digitales. Estos se utilizan en una amplia gama de aplicaciones, desde la formación de simulaciones hasta la visualización arquitectónica y la optimización de productos. La mejora de las características de renderizado y las funciones de un gemelo digital en Unity es esencial para satisfacer las crecientes expectativas de los usuarios y mantener la competitividad en el mercado.

La tecnología en el campo del renderizado y la simulación 3D está evolucionando a un ritmo constante. La búsqueda de la excelencia en la calidad de renderizado puede aprovechar los avances en técnicas de renderizado fotorrealistas, sombreado avanzado, iluminación global y física realista.

La investigación en la mejora de las características de renderizado y funciones de un gemelo digital en Unity puede aportar un valioso conocimiento al campo de la simulación y visualización 3D. Los resultados de este trabajo pueden ser de interés para la industria, contribuyendo a un mejor entendimiento de cómo optimizar y enriquecer las experiencias de gemelos digitales.

En resumen, la motivación de este trabajo reside en su capacidad para abordar cuestiones críticas en el ámbito de la visualización de los gemelos digitales, ofreciendo soluciones relevantes y tecnológicamente avanzadas. La investigación realizada en este proyecto busca ampliar el conocimiento en el campo y contribuir al desarrollo y la mejora de aplicaciones para este tipo de visualizaciones.

1.4. Objetivos

El objetivo de este trabajo es seguir desarrollando el ámbito de la ingeniería a través de la herramienta Unity3D, concretamente el área de visualización de los gemelos digitales, aprovechando toda la potencia de su motor gráfico y haciendo uso de su renderizado en Alta Definición (HD) que, aunque requiere más recursos, propone un modelo más acorde a la realidad.

Específicamente, se trata mejorar el renderizado de un gemelo digital en Unity3D ya existente. Este modelo ha sido seleccionado y proporcionado por el cliente con el fin de alcanzar dichas mejoras.

Una vez realizada la actualización del renderizado, el siguiente paso es adaptar y optimizar la funcionalidad o funcionalidades que sean necesarias y posteriormente, validar que dicha funcionalidad o funcionalidades funcionen correctamente en la visualización con renderizado en Alta Definición.

1.5. Estructura del documento

- **Capítulo 1. Introducción.** A lo largo de este primer capítulo se da una visión general tanto del grupo EDAG como de la empresa EDAG PS y su departamento de Automatización, para el cual se ha desarrollado el proyecto. Asimismo, se exponen la motivación, los objetivos del presente documento y la estructura del mismo.
- **Capítulo 2. Estado del arte: Gemelos Digitales en la Industria 4.0.** Durante el segundo capítulo se explica qué es la Industria 4.0 y el estado en el que se encuentran los gemelos digitales dentro de ella, los diferentes tipos de gemelos digitales que existen y el software que se va a emplear como base, en este caso Unity3D.
- **Capítulo 3. Renderizados de Unity 3D.** En este capítulo se explica qué es el renderizado, los tipos que existen en Unity y se exponen las diferentes características que poseen el renderizado Estándar, el renderizado Universal y el renderizado de Alta Definición. Asimismo, se realiza una comparativa entre los renderizados Universal y de Alta Definición y se expone por qué se elige el renderizado de Alta Definición para realizar la conversión del gemelo digital.
- **Capítulo 4. Entorno del gemelo digital.** En el cuarto capítulo se exponen las diferentes características que posee el gemelo digital con renderizado Estándar que se toma como punto de partida del trabajo, se explica cómo realizar la conversión desde el renderizado Estándar hacia el renderizado de Alta Definición y se realiza una comparativa del gemelo digital con ambos renderizados.
- **Capítulo 5. Funcionalidades del gemelo digital en Alta Definición.** En este quinto capítulo se analizan las diferencias técnicas que contiene el gemelo digital con cada renderizado, se muestran las propuestas para solucionar los problemas con algunas de las funcionalidades y se explica la propuesta definitiva y su validación.
- **Capítulo 6. Conclusiones y trabajo futuro.** En el último capítulo se concluye el proyecto, se elaboran las conclusiones y se enfocan las posibles líneas de trabajo futuras tras la elaboración de este proyecto.

Capítulo 2

Estado del arte: Gemelos digitales en la Industria 4.0

2.1. Industria 4.0

La Industria 4.0 se describe como la integración de tecnologías digitales inteligentes en la esfera de la fabricación y los procesos industriales. Comprende una serie de tecnologías que abarcan desde las redes industriales del Internet de las cosas (IoT) y la Inteligencia Artificial (IA) hasta el Big Data, la robótica y la automatización. La Industria 4.0 posibilita la fabricación inteligente y la creación de fábricas de última generación, con el propósito principal de mejorar la productividad, la eficiencia y la flexibilidad, al tiempo que facilita una toma de decisiones más inteligente y personalizada en las operaciones de fabricación y la cadena de suministro.

Desde principios del siglo XIX, hemos sido testigos de tres importantes revoluciones industriales. El término revolución se aplica debido a que la innovación que las impulsó no se limitó a mejorar ligeramente la productividad y la eficiencia, sino que transformó por completo la forma en que se producían los bienes y se llevaba a cabo el trabajo. Como se muestra en la Figura 2.1, en la actualidad, nos encontramos inmersos en la Cuarta Revolución Industrial, ampliamente conocida como la Industria 4.0. [9]

- **Primera Revolución Industrial.** A finales del siglo XVIII, la Primera Revolución Industrial se inauguró con la invención de la máquina de vapor, lo que redujo la dependencia industrial de la mano de obra humana y animal, marcando el inicio de una nueva era de manufactura y precisión en la ingeniería.
- **Segunda Revolución Industrial.** Un siglo después, el aumento en el uso del petróleo y la electricidad permitió que la maquinaria fuera más ágil y

menos voluminosa. La Segunda Revolución Industrial fue impulsada por la introducción de la línea de ensamblaje y los métodos de producción en masa, muchos de los cuales aún persisten en la actualidad.

- **Tercera Revolución Industrial.** A mediados del siglo XX, las computadoras se convirtieron en una parte integral de la escena industrial. La Tercera Revolución Industrial presenció el surgimiento inicial de la automatización de fábricas y la robótica. En este período, también se utilizaron por primera vez sistemas informáticos empresariales para gestionar y analizar datos.
- **Cuarta Revolución Industrial.** En la actualidad, la manufactura se encuentra cada vez más impulsada por la información, con la recopilación constante de grandes cantidades de datos provenientes de diversas fuentes en todo el mundo, en tiempo real las 24 horas del día. La Inteligencia Artificial desempeña un papel central en la Cuarta Revolución Industrial, permitiendo a los fabricantes no solo recolectar todos esos datos, sino también utilizarlos para su análisis, predicción, comprensión e informes. La Industria 4.0 se caracteriza por la integración fluida de una variedad de sistemas, herramientas e innovaciones.

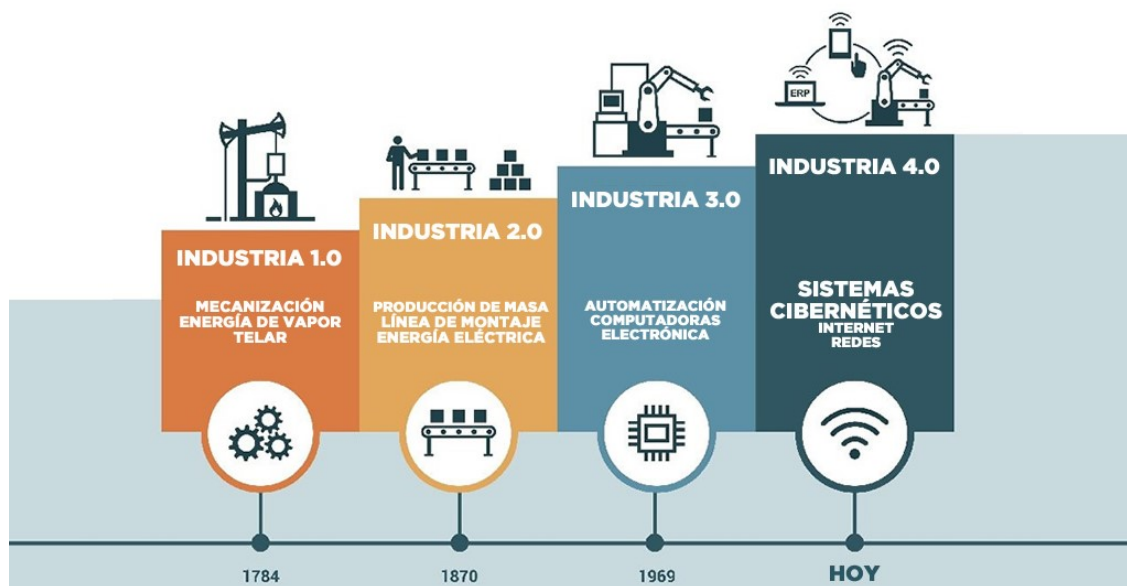


Figura 2.1: De la Industria 1.0 a la industria 4.0. Fuente: Comunidad Virtual Externadista.

2.1.1. Características de la Industria 4.0

La característica sobresaliente de la Industria 4.0 radica en la convergencia de tecnologías como una herramienta para optimizar el tiempo, facilitar la toma de decisiones y minimizar errores. Estas tecnologías previamente mencionadas se distinguen por sus características específicas:

- **Descentralización.** En el contexto de la Industria 4.0, la descentralización implica que las máquinas funcionan de manera autónoma sin depender de la intervención humana. Operan mediante sensores conectados a una red que les permite tomar decisiones automatizadas basadas en datos de rendimiento.
- **Interoperabilidad.** En la Industria 4.0, la interoperabilidad se refiere a la capacidad de diversos sistemas, ya sean físicos, humanos o informáticos, para comunicarse eficazmente entre sí. Un ejemplo claro de esto son las páginas web, que son accesibles a través de distintos navegadores gracias a estándares abiertos que facilitan el intercambio de datos.
- **Virtualización.** La virtualización permite la distribución de las funciones de una máquina física entre múltiples usuarios o entornos, lo que maximiza la utilización de la capacidad de la máquina y su versatilidad.
- **Respuesta en tiempo real.** Las tecnologías de la Industria 4.0 se destacan por la entrega instantánea de datos y la ejecución de algoritmos en tiempo real. Esto posibilita obtener resultados de forma inmediata y aborda cualquier anomalía antes de que esta afecte negativamente el proceso de producción.
- **Modularidad.** La modularidad, mediante la conexión y desconexión de diferentes módulos, permite a las empresas fabricar productos en secuencia sin necesidad de reconfigurar toda la línea de ensamblaje. Esto agiliza el proceso de producción y mejora la flexibilidad.

Estas características son esenciales para entender cómo la Industria 4.0 revoluciona la forma en que las empresas operan y cómo se están aprovechando las tecnologías para lograr eficiencia y competitividad en la era de la automatización y la digitalización.[10]

2.1.2. Tecnologías de la Industria 4.0

La Industria 4.0 se sustenta en una serie de nueve pilares tecnológicos que establecen la base para la creación de sistemas inteligentes y autónomos, permitiendo una transformación significativa en la forma en que las empresas operan. Si bien muchas de estas tecnologías avanzadas se utilizan por separado en la actualidad, el potencial real de la Industria 4.0 se manifiesta cuando se combinan de manera sinérgica.

- **Big Data y analíticas de IA.** En el contexto de la Industria 4.0, el Big Data se recopila de diversas fuentes, incluyendo activos, equipos e IoT. Estos datos no se limitan a la fábrica, sino que se extienden a áreas como las revisiones de clientes, tendencias de mercado e información meteorológica, lo que informa la toma de decisiones en áreas como I+D y diseño.

- **Integración horizontal y vertical.** La Industria 4.0 se caracteriza por una integración estrecha tanto a nivel de campo, donde los procesos están conectados en toda la cadena de producción y suministro, como a nivel organizativo, donde los datos fluyen sin obstáculos entre la planta de fabricación y otras áreas empresariales.
- **Computación en la Nube.** La computación en la nube desempeña un papel fundamental en la Industria 4.0, al proporcionar la base tecnológica para la Inteligencia Artificial, el aprendizaje automático y la integración del IoT. Los datos que alimentan estas tecnologías residen en la nube, permitiendo una comunicación en tiempo real entre sistemas ciberfísicos.
- **Realidad Aumentada (RA).** La realidad aumentada superpone información digital en un entorno real, lo que permite a los trabajadores visualizar datos de IoT, instrucciones de montaje, contenido de capacitación y más a medida que observan elementos físicos, como productos o equipos.
- **Internet de las cosas industrial (IIoT).** El IIoT es central en la Industria 4.0 y proporciona datos en tiempo real sobre el estado, rendimiento y ubicación de dispositivos, maquinaria y productos, lo que permite una cadena de suministro más eficiente y una toma de decisiones más informada.
- **Fabricación aditiva e impresión 3D.** La impresión 3D ha evolucionado desde su origen como una herramienta de prototipado rápido hacia aplicaciones más amplias, como la fabricación distribuida y la personalización en masa.
- **Robots autónomos.** La Industria 4.0 ha dado lugar a una nueva generación de robots autónomos capaces de realizar tareas con mínima intervención humana. Equipados con tecnologías avanzadas, estos robots pueden llevar a cabo tareas complejas y sensibles, además de interactuar con su entorno.
- **Gemelos digitales.** Los gemelos digitales, basados en datos de sensores IoT, permiten a las empresas comprender y mejorar el rendimiento y el mantenimiento de sistemas e productos industriales. Estas simulaciones virtuales son esenciales para la toma de decisiones informadas. Asimismo, sobre este apartado se centrará el presente proyecto.
- **Ciberseguridad.** Dada la creciente conectividad y el uso intensivo del Big Data en la Industria 4.0, la ciberseguridad se convierte en una prioridad. Las empresas deben implementar medidas de seguridad avanzadas, como la arquitectura de redes de confianza cero (Zero Trust) y tecnologías como el aprendizaje automático y la cadena de bloques (blockchain), para proteger sus datos y operaciones.



Figura 2.2: Industria 4.0. Fuente: SAP ES.

La combinación de estos pilares tecnológicos, como se muestra en la Figura 2.2, es esencial para desbloquear todo el potencial de la Industria 4.0 y permitir un salto significativo en la eficiencia y la competitividad de las empresas en la era digital. [9]

2.1.3. Objetivos de la Industria 4.0

La Industria 4.0 representa una nueva revolución que fusiona técnicas avanzadas de producción y operaciones con tecnologías inteligentes que se integran en las estructuras organizativas, el personal y los activos. En esta transformación, emergen tecnologías innovadoras como la robótica, la analítica, la inteligencia artificial, las tecnologías cognitivas, la nanotecnología y el Internet de las Cosas, entre otras. Las empresas deben discernir cuáles de estas tecnologías satisfacen de mejor manera sus necesidades antes de comprometer inversiones. Aquellas organizaciones que no comprendan los cambios y las oportunidades que trae consigo la Industria 4.0 corren el riesgo de perder participación en el mercado.

Para los líderes tradicionales, quienes están acostumbrados a lidiar con datos y procesos lineales, la transición hacia esta nueva revolución industrial, que proporciona acceso en tiempo real a datos y conocimientos empresariales, supondrá una redefinición de la forma en que conducen sus operaciones. La integración digital de información proveniente de diversas fuentes y ubicaciones posibilita un flujo ininterrumpido de actividades comerciales. Este ciclo continuo se fundamenta en el acceso en tiempo real a datos, impulsado por la interacción constante entre los entornos físicos y digitales. Este flujo se concreta mediante una serie de etapas iterativas conocidas como de lo físico a lo digital y de vuelta a lo físico (PDP).

- **Del mundo físico al digital.** En esta fase, se captura información del mundo físico y se genera un registro digital de la misma.
- **De lo digital a lo digital.** En esta etapa, la información se comparte y se interpreta mediante el uso de análisis avanzados, escenarios analíticos y técnicas de inteligencia artificial con el fin de desvelar información relevante.
- **Del mundo digital al físico.** Se aplican algoritmos para traducir decisiones tomadas en el mundo digital en acciones efectivas, estimulando cambios y acciones en el mundo físico.

Esta transformación persigue optimizar la eficiencia, impulsar la innovación y proporcionar una ventaja competitiva en un entorno empresarial en constante evolución. [10]

2.1.4. Soluciones gracias a la Industria 4.0

La transformación digital asociada a la Industria 4.0 implica una evolución tanto en términos culturales como operativos. Al unificar y conectar personas, datos y activos, se abren posibilidades prácticamente ilimitadas para optimizar diversos aspectos. A continuación, se describen ejemplos de cómo las soluciones relacionadas con la Industria 4.0 están contribuyendo a mejorar la eficiencia, visibilidad y sostenibilidad en el entorno de la fabricación y las cadenas de suministro. [9]

- **Plataformas de diseño colaborativo.** La Industria 4.0 respalda la creación de plataformas de diseño colaborativo donde equipos de I+D, diseñadores de productos y otras partes interesadas en la organización pueden acceder y contribuir a la información estratégica y de diseño. Este enfoque colaborativo promueve la interacción entre diferentes funciones, el intercambio de conocimientos y la toma de decisiones basadas en datos, lo que resulta en el desarrollo ágil de productos centrados en las necesidades del cliente.
- **Mantenimiento predictivo.** La integración de sensores de IoT y análisis de datos permite a las empresas monitorear el estado de sus equipos en tiempo real. Los algoritmos de mantenimiento predictivo identifican posibles fallos antes de que ocurran, lo que posibilita la implementación de acciones proactivas para reducir el tiempo de inactividad y prolongar la vida útil de los activos.
- **Optimización de la cadena de suministro.** La Industria 4.0 habilita la visibilidad completa a lo largo de la cadena de suministro global. Con datos en tiempo real que abarcan desde proveedores hasta niveles de inventario, programas de producción, demanda del cliente y operaciones internas, es posible optimizar la logística, equilibrar la oferta y la demanda, mejorar la eficiencia en la gestión de pedidos y, en general, potenciar la eficiencia en la cadena de suministro y la fabricación.

- **Fabricación ágil.** La inteligencia artificial y análisis avanzados permiten la recopilación y análisis en tiempo real de información estratégica y retroalimentación de los clientes procedente de fuentes como redes sociales, revisiones en línea y el soporte al cliente. Los equipos de I+D y diseñadores de productos pueden utilizar estos datos para identificar las preferencias del consumidor, áreas de mejora y tendencias emergentes, lo que facilita la rápida adaptación de productos a las demandas del mercado y promueve la satisfacción y fidelización de los clientes.
- **Control de calidad y detección de defectos.** Mediante el aprovechamiento de dispositivos IoT y algoritmos de aprendizaje automático (machine learning), es posible recopilar datos en tiempo real de todas las líneas de producción. La monitorización continua del proceso de fabricación permite detectar anomalías, identificar problemas de calidad y tomar medidas correctivas de manera oportuna, garantizando un control efectivo de la calidad del producto.
- **Prácticas de economía circular.** La Industria 4.0 facilita la implementación de prácticas de economía circular, centradas en la reducción de residuos y la maximización de la reutilización, renovación y reciclaje de materiales. El seguimiento de los ciclos de vida de los productos, la implementación de la logística inversa para devoluciones y la optimización de la recuperación de recursos valiosos son posibles mediante el uso de análisis de Big Data y redes de IoT. Además, el uso de analítica avanzada respaldada por la IA ayuda a los diseñadores de productos a crear productos más sostenibles desde el principio, minimizando el desperdicio de recursos y facilitando su reciclaje o reutilización.
- **Monitoreo y optimización de la huella de carbono.** Las tecnologías de la Industria 4.0 permiten la recopilación y análisis en tiempo real de datos relacionados con el consumo de energía, emisiones de transporte y otros factores que contribuyen a la huella de carbono de una empresa. La medición precisa de las emisiones permite identificar áreas de mejora, implementar medidas de eficiencia energética y desarrollar estrategias para reducir la huella de carbono en general, lo que facilita el cumplimiento de objetivos de sostenibilidad cada vez más ambiciosos.

2.2. Gemelos digitales

2.2.1. Introducción a los gemelos digitales

Los gemelos digitales, también conocidos como digital twins, representan un modelo virtual que replica con precisión un objeto físico, proceso o sistema. Estos modelos virtuales se emplean para llevar a cabo simulaciones y analizar el comportamiento de un producto digital, con el propósito de ajustar eficazmente las soluciones en el producto físico real [11]. Una vez que los datos han sido procesados, el modelo virtual se convierte en una herramienta para realizar simulaciones, investigar cuestiones de rendimiento y proponer mejoras potenciales. El enfoque principal es generar información valiosa que pueda ser posteriormente aplicada al objeto físico original. [12]

Los gemelos digitales se enriquecen con datos en tiempo real, los cuales pueden ser recolectados a través de sensores o tecnologías de análisis de datos. Esta tecnología integra componentes como el internet de las cosas, la inteligencia artificial, el aprendizaje automático y el análisis de datos. [11]

Estas réplicas virtuales brindan la capacidad de analizar procesos reales, experimentar con objetos físicos existentes y crear escenarios hipotéticos para predecir comportamientos. La creación de un gemelo digital puede abordar una amplia variedad de objetos, desde motores de avión hasta plantas de producción o incluso ciudades enteras [11]. La tecnología de gemelos digitales permite supervisar el rendimiento de un activo, identificar posibles fallas y tomar decisiones más fundamentadas en lo que respecta al mantenimiento y al ciclo de vida del objeto en cuestión. [13]

Un gemelo digital se define como un modelo virtual meticulosamente diseñado para replicar un objeto físico en detalle. El objeto de interés, como se ilustra en la Figura 2.3, incorpora una serie de sensores que abarcan áreas críticas de funcionamiento. Estos sensores recopilan datos relacionados con diversos aspectos del desempeño del objeto físico, tales como producción de energía, temperatura, condiciones climáticas, y otros parámetros relevantes. Estos datos son posteriormente transferidos a un sistema de procesamiento y aplicados al modelo digital correspondiente.

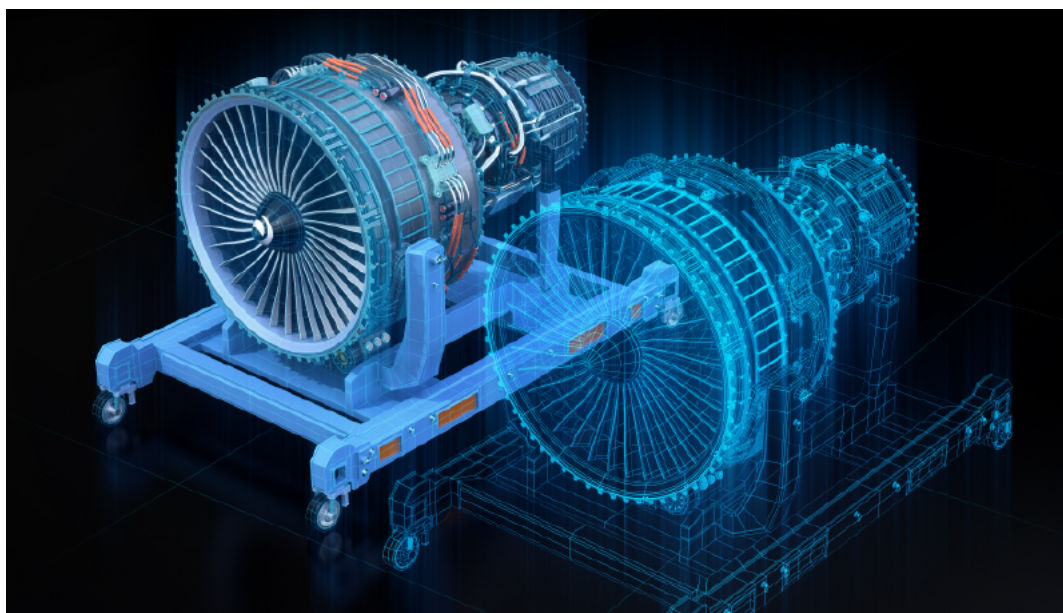


Figura 2.3: Ejemplo de un componente real y su gemelo digital. Fuente: ITAInnova.

Una vez que los datos han sido procesados, el modelo virtual se convierte en una herramienta que permite realizar simulaciones, abordar problemas de rendimiento y proponer mejoras potenciales. El propósito central consiste en generar información valiosa que pueda ser aplicada de manera efectiva al objeto físico original. [12]

Los gemelos digitales se componen principalmente de tres elementos: el objeto físico, su contraparte virtual y las interconexiones que se establecen entre ambos. La calidad y el nivel de detalle de la versión digital resultante dependen directamente de la cantidad de datos utilizados en su creación y actualización. [11]

2.2.2. Características de los gemelos digitales

El gemelo digital en la industria 4.0 es viable gracias a una serie de características clave:

- **Conectividad.** En primer lugar, el gemelo digital se encuentra conectado a su contraparte física. Esta conexión es un elemento fundamental de la tecnología de gemelos digitales y se logra mediante sensores ubicados en el objeto físico y su entorno. Los datos recopilados se transmiten a un ordenador que alberga el gemelo digital.
- **Terminología.** Los gemelos digitales han propiciado la diversificación de la industria al permitir que el sector secundario de la economía ofrezca servicios como el mantenimiento, ingresando así en el sector terciario.

- **Homogeneización de datos.** Los gemelos digitales contribuyen a la homogeneización de datos. Al digitalizar y almacenar información de productos en forma de gemelos digitales, se logra un desacoplamiento entre la información y su parte física. Esto facilita el almacenamiento digital de información detallada sobre un producto.
- **Eficiencia en el tratamiento de datos.** El procesamiento de datos digitales es más rentable, y la ley de Moore prevé un aumento exponencial en la potencia de cálculo con costes decrecientes. Esto reduce los costes marginales en el desarrollo y el uso de gemelos digitales, haciéndolos más asequibles y efectivos para pruebas, predicciones y resolución de problemas en comparación con los modelos físicos.
- **Experiencia de usuario convergente.** La homogeneización y el desacoplamiento de la información conducen a una convergencia en la experiencia del usuario. La información digitalizada permite explorar diferentes aspectos de un objeto físico en el mundo virtual. Los gemelos digitales permiten que múltiples agentes compartan información detallada sobre un objeto sin restricciones de tiempo o ubicación.
- **Reprogramación e inteligencia.** Dado que el gemelo digital es un sistema informático, puede ser reprogramado. Esta característica permite la incorporación de nuevas funcionalidades y la mejora de productos al analizar el rendimiento de la parte física bajo diferentes configuraciones.
- **Rastros digitales.** Los gemelos digitales dejan rastros digitales, lo que facilita el diagnóstico de problemas. Los ingenieros pueden analizar estos rastros para identificar discrepancias entre el comportamiento de la real y el digital, lo que ayuda en el proceso de diagnóstico y mejora del diseño.
- **Modularidad.** La tecnología de gemelos digitales permite la modularidad, es decir, la capacidad de dividir un producto en partes que pueden tratarse de forma independiente. Esta modularidad permite realizar mejoras y cambios específicos en los componentes del producto sin afectar el conjunto. [14]

2.2.3. Tipos de gemelos digitales

Hay tres categorías de gemelos digitales que varían según la fase de producción del producto.

- **Prototipo de gemelo digital (DTP).** Este tipo de gemelo digital se desarrolla cuando el producto aún no ha sido fabricado físicamente. Un prototipo virtual se crea para proporcionar un análisis detallado de cómo se vería y se comportaría el producto físico.

- **Instancia de gemelo digital (DTI).** Cuando el producto real ya existe, se crea una instancia de gemelo digital para llevar a cabo pruebas en diversos entornos en los que el producto podría ser aplicado.
- **Agregado de gemelos digitales (DTA).** En esta categoría, se aprovecha la información y los datos previamente recopilados de un producto real para predecir y evaluar sus capacidades y rendimiento.

Asimismo, hay varios tipos de gemelos digitales, que con frecuencia se pueden ejecutar en paralelo dentro del mismo sistema. Si bien algunos gemelos digitales replican solo partes individuales de un objeto, todos son fundamentales para proporcionar una representación virtual. Los tipos más comunes de gemelos digitales son los siguientes. [15]

- **Gemelos digitales de productos.** Los gemelos digitales de productos son esenciales en el ciclo de desarrollo y fabricación. Al crear una réplica virtual precisa de un producto, desde su diseño conceptual hasta su fabricación y uso final, las empresas pueden obtener una visión completa de su rendimiento antes de que se fabrique físicamente. Esto elimina la necesidad de crear múltiples prototipos costosos y largos procesos de ensayo y error.

Estos gemelos permiten simular el comportamiento del producto en diversas condiciones y ajustar su diseño y características en un entorno virtual. Los datos recopilados durante el uso del producto real se pueden comparar con el gemelo digital para mejorar su rendimiento y eficiencia a lo largo de su ciclo de vida.

- **Gemelos digitales de producción.** En el ámbito de la fabricación, los gemelos digitales desempeñan un papel crucial en la optimización de los procesos de producción. Antes de implementar cambios en una planta de fabricación o introducir nuevos equipos, es posible simular su funcionamiento en un entorno digital.

Esto permite a las empresas evaluar cómo se comportarán los cambios en diferentes condiciones y escenarios. Igualmente, al crear gemelos digitales de todos los equipos de fabricación, se pueden prevenir costosos tiempos de inactividad al anticipar y programar el mantenimiento preventivo en función de datos precisos.

Es este tipo de Gemelos digitales sobre el que se va a basar la redacción de este proyecto debido al interés en desarrollar el desempeño de una planta de producción y a su vez no estar obligados a parar la producción mientras se está inmerso en la búsqueda de las mismas.

- **Gemelos digitales de rendimiento.** Los gemelos digitales de rendimiento se centran en la recopilación y análisis de datos en tiempo real generados por productos y plantas en funcionamiento. Esto es esencial para tomar decisiones informadas y mejorar la eficiencia operativa.

Al analizar estos datos, las empresas pueden descubrir nuevas oportunidades de negocio, mejorar sus modelos virtuales existentes y ajustar las operaciones para lograr una mayor eficiencia. Incluso estos gemelos permiten la captura y agregación de datos operativos clave, lo que resulta en una toma de decisiones más precisa y una mejora continua de los productos y procesos de producción.

- **Duplicado de activos.** Cuando se trabaja con sistemas complejos, la interacción entre los componentes es esencial para comprender su funcionamiento y desempeño. Los gemelos digitales de activos desempeñan un papel crucial al permitirnos estudiar esta interacción de manera detallada. Esto implica que podemos observar cómo varios componentes trabajan juntos en un sistema, comprendiendo mejor su comportamiento y desempeño en conjunto.

A medida que se estudia esta interacción, se generan una gran cantidad de datos de rendimiento. Estos datos capturan cómo los componentes influyen entre sí y cómo se comportan en diferentes situaciones. Luego, estos datos se procesan y analizan, lo que permite extraer información valiosa y conocimientos útiles. [12]

- **Gemelos del sistema.** Estos modelos virtuales permiten comprender y optimizar el funcionamiento de sistemas completos, en lugar de limitarse a componentes individuales. Al representar cómo diversos activos interactúan en conjunto como una unidad funcional, los gemelos del sistema ofrecen una visión globalizada de un proceso o sistema.

La utilidad de los gemelos del sistema radica en su capacidad para simular y analizar la dinámica y el rendimiento de sistemas complejos en un entorno virtual. Esto resulta beneficioso en diversas aplicaciones, desde la gestión de la cadena de suministro y la planificación de la producción hasta la optimización de procesos industriales. Los datos generados por los gemelos del sistema pueden ayudar a identificar cuellos de botella, mejorar la eficiencia y tomar decisiones informadas para maximizar el rendimiento.

2.2.4. Ventajas que aportan los gemelos digitales

Las principales ventajas de la tecnología digital twins se exponen a continuación.

- **Optimización de procesos.** La creación de réplicas digitales posibilita acelerar el proceso de producción de bienes y sistemas incluso antes de su existencia física. Al simular distintos escenarios, se visualiza la respuesta de los productos o instalaciones ante posibles fallos, permitiendo realizar ajustes pertinentes antes de la fase de producción efectiva.
- **Reducción de costes.** La obtención de una eficiencia óptima en los procesos productivos conlleva una variedad de beneficios directos, incluyendo la reducción de costos de producción de múltiples maneras: desde el mejoramiento de

la eficiencia energética en las líneas hasta el aumento de la producción por unidad, la minimización de tiempos improductivos y la prevención de posibles averías, entre otros aspectos.

- **Capacidades predictivas.** Los gemelos digitales poseen la capacidad de proporcionar una representación exhaustiva y digital de plantas de fabricación, edificios comerciales o instalaciones, inclusive si constan de miles de componentes. A través de sensores inteligentes que monitorean el desempeño de cada elemento, se identifican y señalan problemas o fallos a medida que surgen. Esto permite tomar acciones preventivas ante las primeras señales de inconvenientes en lugar de aguardar a que el equipo falle por completo.
- **Flexibilidad.** La evolución de los mercados, la automatización de procesos y el avance de nuevas tecnologías han generado una transformación en los entornos industriales. Conceptos como la *Fábrica Flexible*, que implica contar con líneas de producción adaptables a diversos formatos y referencias de manera ágil, son ejemplos de esta evolución. En este contexto, el gemelo digital se posiciona como una herramienta sobresaliente para optimizar la producción en estos escenarios dinámicos.
- **Sostenibilidad.** El gemelo digital aporta una serie de ventajas relacionadas con la sostenibilidad. Evita gastos en la construcción de prototipos costosos y previene decisiones poco acertadas, como la elección de sistemas de transporte automatizados. Además, al optimizar procesos y almacenes, las empresas pueden reducir de forma notable su consumo energético y preservar valiosos recursos.
- **Mejor rendimiento.** La información en tiempo real brindada por los gemelos digitales posibilita la optimización del desempeño de equipos, plantas e instalaciones. Esta información permite abordar problemas tan pronto como surgen, asegurando el funcionamiento óptimo de los sistemas y disminuyendo el tiempo de inactividad.
- **Supervisión remota.** La naturaleza virtual de los gemelos digitales habilita la supervisión y el control remoto de instalaciones. Esta capacidad de supervisión a distancia también disminuye la necesidad de personal en la vigilancia de equipos industriales que podrían resultar peligrosos.

Aunque las simulaciones y los gemelos digitales emplean modelos digitales para replicar diversos procesos de un sistema, la diferencia fundamental radica en que un gemelo digital constituye un entorno virtual mucho más valioso para la investigación. Más allá de la escala, donde una simulación típicamente se centra en un proceso particular, un gemelo digital tiene la capacidad de llevar a cabo múltiples simulaciones y estudiar diversos procesos de manera autónoma.

Sin embargo, las discrepancias no se limitan a eso. Por ejemplo, las simulaciones suelen carecer de datos en tiempo real, mientras que los gemelos digitales se

fundamentan en un flujo constante de información bidireccional. Esta interacción se inicia cuando los sensores del objeto aportan datos al procesador del sistema y, posteriormente, los conocimientos generados por el procesador son devueltos al objeto original. [16]

La riqueza de datos actualizados constantemente y su amplio alcance en distintos ámbitos, sumados al poder computacional adicional que ofrece su entorno virtual, permiten a los gemelos digitales abordar una gama más amplia de cuestiones desde múltiples perspectivas en comparación con las simulaciones convencionales. Esta capacidad mejorada no solo potencia la investigación, sino que también amplía el potencial para la mejora de productos y procesos. [12]

2.2.5. Aplicaciones de los gemelos digitales

Las herramientas de gemelos digitales han encontrado aplicación en una variedad de sectores, desde la industria hasta la agricultura, la atención médica y más. A continuación, se presenta un resumen general de sus aplicaciones:

- **Operaciones de fabricación.** Los gemelos digitales se han vuelto fundamentales en la respuesta a los desafíos industriales actuales. Estos modelos virtuales fomentan la fabricación inteligente al simular decisiones y optimizar procesos.
- **Servicios de atención médica.** Al igual que se perfilan productos con gemelos digitales, también se pueden perfilar los pacientes que reciben servicios de atención médica. Ejemplos incluyen la medicina personalizada y la creación de gemelos digitales para estudiar corazones fetales.
- **Equipos de generación de energía.** Motores grandes y turbinas de generación de energía encuentran beneficios sustanciales al emplear gemelos digitales para establecer cronogramas de mantenimiento.
- **Estructuras y sistemas.** Desde grandes edificios hasta sistemas operativos dentro de estas estructuras, como los sistemas HVAC, pueden mejorarse a través de gemelos digitales, especialmente durante la fase de diseño.
- **Industria automotriz.** Los gemelos digitales se utilizan ampliamente en el diseño de automóviles para mejorar su rendimiento y eficiencia en la producción.
- **Urbanismo.** Ingenieros civiles y arquitectos se benefician del uso de gemelos digitales para planificación urbana, mostrando datos espaciales en tiempo real y creando gemelos digitales de ciudades enteras.

- **Gestión y mantenimiento de infraestructuras.** Los gemelos digitales se emplean para gestionar y mantener infraestructuras públicas y privadas, integrando modelado de información de construcción (BIM) con sistemas de información geográfica.
- **Agricultura y granjas inteligentes.** Los gemelos digitales permiten a los productores gestionar sus operaciones de forma remota, actuando con información en tiempo real y simulando intervenciones.
- **Economía circular.** Se destaca la contribución de los gemelos digitales a la circularidad de productos y la gestión de cadenas de suministro circulares.
- **Piscicultura inteligente.** Se propone el uso de gemelos digitales para la gestión de piscifactorías, integrando sensores, Big Data e IoT para optimizar la gestión.

Estos ejemplos reflejan la versatilidad de los gemelos digitales y su impacto en diversos campos, desde la industria hasta la agricultura, la atención médica y la gestión urbana. [17]

2.3. Unity 3D

Unity es un motor de juego multiplataforma desarrollado por Unity Technologies, cuyo anuncio y lanzamiento inicial se efectuaron en junio de 2005 durante la Conferencia Mundial de Desarrolladores de Apple como un motor de juego destinado al sistema operativo Mac OS X. Desde entonces, el motor ha experimentado un desarrollo gradual y continuo para ofrecer compatibilidad con diversas plataformas, abarcando entornos de escritorio, dispositivos móviles, consolas y sistemas de realidad virtual. Su notable popularidad se observa especialmente en el ámbito del desarrollo de juegos para dispositivos móviles con sistemas iOS y Android. Cabe destacar que se le considera de uso accesible para desarrolladores principiantes y goza de reconocimiento en el campo del desarrollo de juegos independientes.

El alcance de este motor no se limita exclusivamente a la creación de juegos en dos y tres dimensiones, sino que también permite la generación de simulaciones interactivas y otras experiencias. Además, es importante señalar que su adopción trasciende el ámbito de los videojuegos, extendiéndose a industrias diversas como la cinematográfica, la automotriz, la arquitectura, la ingeniería, la construcción e incluso las Fuerzas Armadas de los Estados Unidos.

Unity brinda a los usuarios la capacidad de crear juegos y experiencias, tanto en dos como en tres dimensiones. El motor de este sistema proporciona una interfaz de programación de aplicaciones (API) principal en C#. Esta API se extiende tanto

al editor de Unity en forma de complementos como a los juegos en sí, incorporando además funcionalidades de arrastrar y soltar. [18]

2.3.1. Servicios de Unity

Dentro del contexto de los juegos en dos dimensiones, Unity dispone de un motor avanzado para la representación de mundos en 2D. En lo que respecta a los juegos en tres dimensiones, Unity concede la especificación de compresión de texturas, mapas de textura en múltiples niveles (mipmaps) y ajustes de resolución, teniendo en cuenta cada plataforma compatible con el motor de juego. Además, proporciona soporte para el mapeo de relieve, mapeo de reflexión, mapeo de paralaje, oclusión ambiental en espacio de pantalla (SSAO), sombras dinámicas mediante el uso de mapas de sombras, representación en textura y efectos de postprocesamiento a pantalla completa.

Este sistema ofrece tres tipos de renderizado por separado: el renderizado Estándar o Built-in, el renderizado Universal y el renderizado en Alta Definición. Es importante destacar que las tres opciones de renderizado son incompatibles entre sí. Unity dispone de una herramienta para actualizar los sombreadores (shaders) del renderizado Estándar a los renderizados Universal o de Alta definición.

Los creadores pueden desarrollar y comercializar activos generados por los usuarios a otros creadores de juegos a través de la tienda de activos de Unity. Esto incluye activos en 3D y 2D, así como entornos que los desarrolladores pueden adquirir y vender.

Unity es un motor multiplataforma. El editor de Unity es compatible con Windows, macOS y la plataforma Linux, mientras que el motor en sí actualmente admite la creación de juegos para más de 19 plataformas diferentes, incluyendo móviles, ordenadores, consolas y realidad virtual. Unity 2020 LTS admite plataformas: móviles, ordenadores, web, de consolas y de realidad virtual o ampliada. [19]

En el ámbito industrial también está presente como se muestra en la Figura 2.4 los fabricantes de automóviles utilizan la tecnología de Unity para crear modelos a escala real de nuevos vehículos en realidad virtual, construir líneas de ensamblaje virtuales y capacitar a los trabajadores. El motor de Unity es utilizado por DeepMind, una empresa de Alphabet Inc., para entrenar inteligencia artificial. Unity Technologies también está explorando otros usos en áreas como arquitectura, ingeniería y construcción. [20]

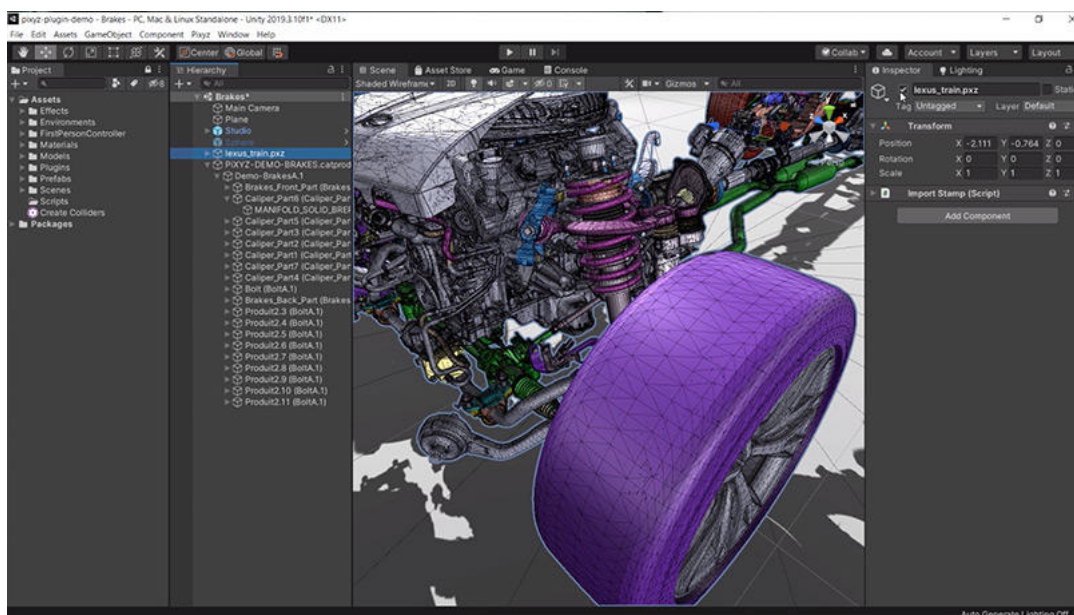


Figura 2.4: Ejemplos de uso de Unity en el ámbito industrial. Fuente: Unity Industrial.

Tienda de activos de Unity

La tienda de activos de Unity o Unity Asset Store emerge como un recurso inestimable para desarrolladores y creadores, proporcionando un amplio repertorio de activos, desde modelos 3D hasta scripts, que pueden potenciar significativamente la calidad y eficiencia de los proyectos.

Los activos de la Unity Asset Store no solo añaden elementos visuales, sino que también pueden optimizar los flujos de trabajo. Scripts, complementos (plugins) y herramientas específicas pueden acelerar tareas recurrentes y simplificar procesos, permitiendo a los desarrolladores centrarse en la creatividad y la innovación. La adquisición de estos activos preexistentes puede ahorrar tiempo y recursos significativos. En lugar de crear elementos desde cero, los desarrolladores pueden centrarse en aspectos más únicos y personalizados del proyecto, aumentando la eficiencia y acelerando el tiempo de desarrollo.

Muchos activos en la tienda de activos de Unity reciben actualizaciones regulares y cuentan con soporte de la comunidad o de los propios creadores. Esto garantiza que los recursos utilizados estén siempre actualizados y puedan adaptarse a las evoluciones del entorno de desarrollo. Al conectarse con la comunidad, los desarrolladores pueden compartir experiencias, aprender nuevas técnicas y encontrar soluciones a desafíos específicos. [21]

2.3.2. Renderizadores de malla en Unity

El renderizador de malla o mesh renderer en Unity es un componente crucial dentro del entorno de este motor de juego que desempeña un papel esencial en la representación visual de objetos 3D en la escena.

En Unity, un objeto 3D generalmente está compuesto por dos elementos principales: una malla o mesh y un material. La malla define la geometría tridimensional del objeto, es decir, su forma y estructura, mientras que el material define cómo se verá ese objeto, especificando detalles como el color, la textura y las propiedades de reflectividad.

El mesh renderer es el componente encargado de tomar esta información y renderizar el objeto en la pantalla. [22] A continuación, se desglosan sus funciones y características principales:

- **Asociación con la malla y el material.**

El renderizado de malla está estrechamente vinculado a la malla y al material de un objeto. A través de este componente, Unity sabe qué forma darle al objeto y cómo presentar su apariencia visual.

- **Renderización en tiempo real.**

Una de las características más destacadas del mesh renderer es su capacidad para renderizar objetos en tiempo real durante la ejecución del juego. Esto significa que los cambios en la posición, rotación, escala o material de un objeto se reflejan de inmediato en la pantalla.

- **Control de visibilidad**

El mesh renderer permite controlar la visibilidad de un objeto en la escena. Puede ser activado o desactivado según sea necesario, lo que resulta útil para optimizar el rendimiento al no renderizar objetos que no están actualmente visibles.

- **Gestión de sombras y luces**

Este componente también juega un papel crucial en la generación de sombras y en la interacción con fuentes de luz en la escena. Esto contribuye a la creación de escenas visualmente realistas al simular cómo la luz interactúa con los objetos.

■ Instancias y representación por lotes

Unity optimiza la renderización de objetos mediante técnicas como la representación por lotes, que agrupa objetos similares para reducir el costo computacional. El renderizado de malla juega un papel importante en este proceso al permitir la renderización eficiente de múltiples instancias del mismo objeto.

En resumen, el renderizado de malla actúa como un componente puente entre la geometría y la apariencia visual de un objeto 3D.

2.3.3. Sombreadores en Unity

Los sombreadores o shaders son la forma en que los ordenadores pueden representar gráficos 3D en 2D. Son esenciales para la mayoría de los videojuegos modernos o simulaciones y un elemento clave para crear imágenes, ya sea para mostrar realmente la escena en 3D o para crear efectos visuales llamativos que den vida al entorno. Gracias a su estructura paralela, también pueden aprovechar las últimas arquitecturas basadas en GPU y ser increíblemente rápidos. Una vez que se logra configurar todo correctamente, los resultados son bastante asombrosos.

Los shaders pueden ser conceptualizados como interfaces de bajo nivel en el ámbito informático. En esencia, son programas computacionales capaces de procesar tanto entradas 2D como 3D y realizar cálculos destinados a la generación de colores para cada píxel en la pantalla. Por lo tanto, se les considera de bajo nivel debido a la manipulación de información de carácter computacionalmente complejo, y de interfaz gráfica debido a su función en la creación de representaciones visuales destinadas a ser observadas y apreciadas por un público. [23]

Es importante destacar que la definición de los sombreadores puede resultar ambigua, variando según la fuente consultada. Desde una perspectiva centrada en el ámbito del juego, un shader se concibe principalmente como un efecto visual que puede ser incorporado en la carpeta de datos de un videojuego para mejorar significativamente aspectos como la iluminación o las texturas del mundo tridimensional. Por otro lado, en comunidades más orientadas hacia la tecnología, se utiliza el término shaders para hacer referencia tanto a código ejecutado en vértices 3D como a instrucciones aplicadas a píxeles e incluso para describir programas más extensos que abarcan tanto código relacionado con entidades tridimensionales como con elementos pixelados. Esta variabilidad en la definición del concepto de sombreador puede resultar de interés para investigaciones futuras en este ámbito.

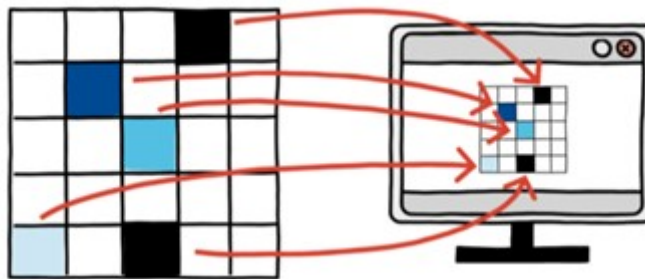


Figura 2.5: Ejemplos de transferencia bits a una pantalla. Fuente: Desconocida.

Suponiendo que se dispone de una imagen y se desea mostrarla en una pantalla como se muestra en la Figura 2.5. Desde una perspectiva técnica, el procedimiento resulta relativamente directo: el ordenador simplemente debe leer el contenido del archivo y realizar una operación de transferencia de bits, comúnmente referida como blit, en la cual se transfiere el color de cada píxel contenido en esta imagen a una posición específica en la pantalla. Estos colores se disponen uno junto al otro de manera secuencial, lo que conduce a la gradual reconstrucción de la imagen y su posterior visualización. Como resultado, se logra de manera exitosa la presentación de la imagen bidimensional en la pantalla.

Sin embargo, si se limitara exclusivamente a la metodología de transferencia píxel a píxel, la concepción visual de la escena tridimensional en el entorno de Unity se volvería inviable. Este escenario se presenta debido a la carencia de píxeles disponibles para su reintegración en la pantalla. El ordenador, se encuentra en posesión exclusiva de un conjunto de coordenadas virtuales tridimensionales que componen la escena. No obstante, el sistema logra generar una representación en tiempo real de estas formas tridimensionales en el monitor, en un proceso análogo al que previamente experimentamos en una imagen bidimensional. Esto es posible gracias a los sombreadores.

Los shaders son fragmentos de código complejos que, en su forma más completa, permiten a una computadora representar un espacio tridimensional en una imagen bidimensional. Además, son capaces de asociar adecuadamente las posiciones en 3D y los datos de renderización, como el color o la rugosidad del material, con el color correcto para el píxel correspondiente en la imagen final. [23]

Los sombreadores son capaces de cerrar la brecha entre la escasa información tridimensional (3D) y la densa información en píxeles bidimensionales (2D), mediante un proceso que consta de tres fases claramente definidas:

- **Sombreador de vértices o Vertex Shader.** En la fase inicial, el sombreador recopila la información sobre la forma necesaria para su renderizado. Esta tarea recae en el sombreador de vértices. Este fragmento de código se ejecuta en paralelo en todos los vértices de la malla 3D y transforma los datos inicia-

les por vértice en datos procesados por vértice. Este proceso conlleva, por lo general, convertir la posición del vértice desde el espacio del objeto al espacio de recorte utilizando la matriz de Modelo-Vista-Proyección. De manera adicional, puede efectuar manipulaciones en la posición del vértice para llevar a cabo deformaciones eficientes de la malla. [24]

- **Rasterización.** El paso siguiente implica la transición desde la representación 3D limitada de la forma, expresada como un conjunto de vértices y triángulos, hacia una retícula mucho más condensada de celdas, los píxeles de la pantalla. Esta fase, conocida como rasterización, es llevada a cabo de forma automática por el ordenador y no requiere la elaboración de un script por parte del desarrollador. Los interpoladores, en esencia, combinan linealmente todas las entradas disponibles y las interpolan para obtener valores intermedios. Al término de la fase de rasterización, se obtiene un conjunto de diminutas celdas 2D que se extienden a lo largo y ancho de toda la pantalla, cada una repleta de un conjunto de propiedades 3D fusionadas: los píxeles o fragmentos. [25]
- **Sombreador de fragmentos o de píxeles.** Se emplean los datos procesados por fragmento para calcular el color real que se exhibirá en pantalla, en la segunda parte del código del sombreador, denominado sombreador de fragmentos. Aquí es donde se puede muestrear una textura de referencia basada en las coordenadas UV de entrada, aplicar algún tono a un color o incluso utilizar los datos de entrada y algunas variables globales para calcular la iluminación. [26]

A continuación, en la Figura 2.6 se presenta un diagrama ilustrativo de todo este proceso, utilizando como ejemplo una pequeña figura con colores distintos en cada vértice:

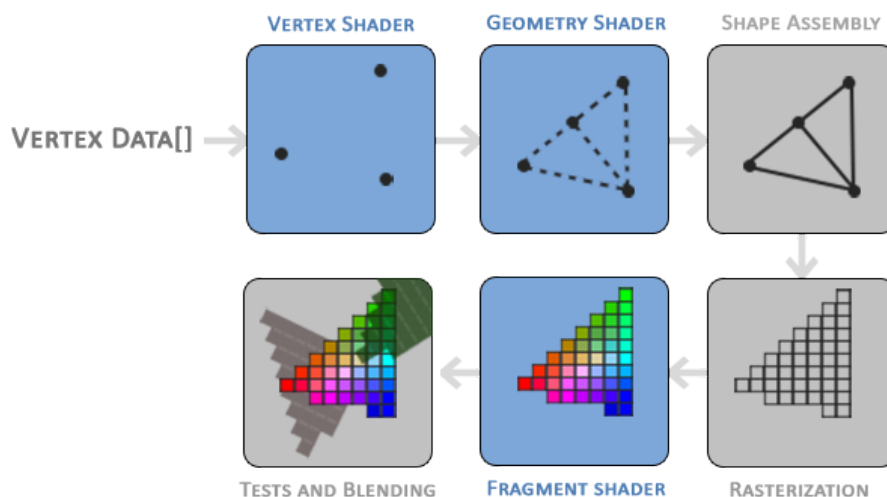


Figura 2.6: Ejemplo de cómo funciona un renderizado típico. Fuente: Learn OpenGL.

Los sombreadores son un tipo de recurso que no puede ser utilizado de manera independiente. Al desarrollar un juego en Unity y buscar la aplicación de un sombreador a un objeto dentro de la escena, es necesario, en realidad, crear un activo intermedio denominado material que depende de dicho sombreador. Luego, se debe configurar la malla del objeto para que utilice este material.

Hay algunos componentes de los sombreadores que se deben tener en cuenta:

Búfer Z

Un componente esencial de un sombreador es cómo interactúa con el entorno en su conjunto, o, dicho de otro modo, qué tipo de comportamiento se espera de un objeto que utiliza el shader.

Esta interacción con la escena se puede separar en dos partes:

Por un lado, es crucial decidir si la cámara debe renderizar el sombreador en la parte frontal o en la parte posterior. Con el fin de optimizar el proceso de renderización, una estrategia inteligente es verificar si algunos objetos opacos están ocultando a otros, y así determinar si podemos ignorar los que están más lejos o descartar todas las caras que no son visibles desde nuestro punto de vista. Para hacerlo, podemos depender de la técnica del Búfer Z.

Por otro lado, aún existen situaciones en las que uno o varios objetos se superponen en el campo visual de la cámara. En dichas circunstancias, es cuando necesario abordar una segunda característica crucial para el shader, el cómo debería mezclarse con los demás elementos. En otras palabras, saber si se requiere que los objetos más cercanos visibles se superpongan a los demás, o si se prefiere tener cierta transparencia y mezclar sus colores de alguna manera. Todos estos aspectos pueden ser definidos a través de la elección del modo de mezcla adecuado para nuestro shader. [27]

Eliminación o culling

Se trata de una propiedad a nivel de polígono que garantiza que, por defecto, las partes de mesh que quedan de espaldas a la cámara sean ignoradas.

La prueba de profundidad, en consecuencia, facilita la optimización de la eficiencia de renderizado al descartar completamente todos los datos que eventualmente terminarían siendo sobrescritos. Dado que se ejecuta entre las etapas del shader de vértices y la rasterización, evita realizar interpolaciones en vértices no utilizados y, en cambio, ayuda al proceso a concentrarse en el objeto visible y las caras en la escena. [28]

2.3.4. Postprocesamiento

El postprocesamiento es un término genérico para un efecto de procesamiento de imagen a pantalla completa que ocurre después de que la cámara dibuje la escena, pero antes de que la escena se represente en la pantalla. El postprocesamiento puede mejorar drásticamente la apariencia visual de un producto con poco tiempo de configuración.

Unity ofrece una gran cantidad de efectos de post procesado y efectos de pantalla completa que puede aumentar considerablemente la apariencia de la escena, como se muestra en la Figura 2.7.



Figura 2.7: Comparación entre imagen sin postprocesado (izquierda) e imagen usando postprocesado (derecha). Fuente: Manual de Unity.

El renderizado Estándar no tiene una técnica de postprocesado integrada como sí la tienen los renderizados universal y de Alta Definición, sino que se tiene que importar un paquete a través del Administrador de paquetes de Unity. [29]

Los efectos que permite añadir la herramienta de postprocesamiento son los siguientes:

- **Oclusión Ambiental.** Se encarga de calcular los puntos en una escena que están expuestos a la iluminación ambiental. Posteriormente, oscurece las áreas que se encuentran ocultas a la luz ambiental, tales como pliegues, agujeros y espacios entre objetos que están cercanos entre sí.
- **Antialiasing.** El efecto de antialiasing tiene como objetivo atenuar la apariencia de los bordes en una escena al rodear dichos bordes con puntos de color similares. Esta técnica reduce el efecto de diente de sierra causado por el aliasing como se muestra en la Figura 2.8.

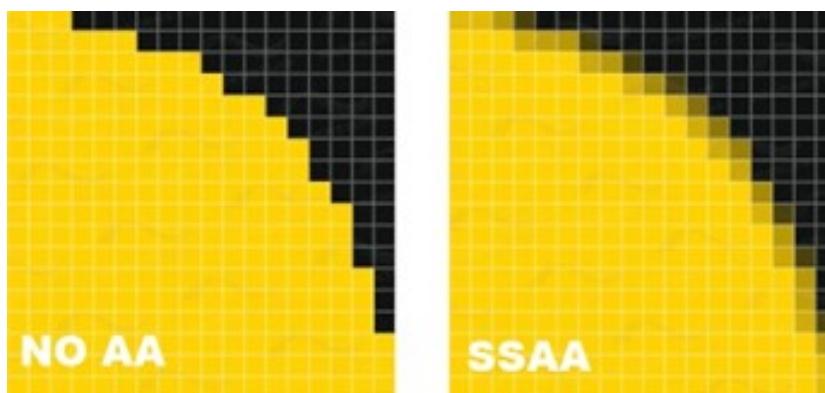


Figura 2.8: Ejemplo de antialiasing suavizando los bordes de las formas. Fuente: Manual de Unity.

- **Autoexposición.** Simula el proceso mediante el cual el ojo humano se adapta en tiempo real a cambios en la luminosidad. Para lograrlo, ajusta de manera dinámica la exposición de una imagen para que se corresponda con su valor de luminancia promedio. En Unity, este efecto genera un histograma en cada fotograma y lo filtra para encontrar el valor promedio de luminancia.
- **Efecto resplandor o Bloom.** Se encarga de producir la irradiación luminosa en áreas brillantes de una imagen. Para lograrlo, genera halos luminosos que se extienden desde las áreas destacadas de la imagen, simulando el fenómeno que ocurre en cámaras del mundo real cuando la intensidad lumínica supera los límites del objetivo.
- **Aberración cromática.** Divide los colores a lo largo de los límites de una imagen en sus canales de rojo, verde y azul. Esto reproduce el efecto que una cámara del mundo real produce cuando la luz se refracta y hace que las longitudes de onda se dispersen en la lente.
- **Mapeo de tonos.** Remapea colores de alto rango dinámico (HDR) en un rango adecuado para medios con un rango dinámico bajo (LDR). Su propósito más común es hacer que una imagen con un rango dinámico bajo parezca tener un rango de colores más amplio. Este resultado aumenta el rango de colores y el contraste en una imagen para dar un efecto más dinámico y realista.
- **Efecto niebla diferida** . Permiten ajustar rangos específicos de tono, saturación o luminosidad.
- **Profundidad de Campo.** Desenfoca el fondo de la imagen mientras los objetos en primer plano permanecen enfocados. Esto simula las propiedades de enfoque de una lente de cámara del mundo real.
- **Viñeta.** Oscurece los bordes de una imagen. Esto simula el efecto en una lente de cámara del mundo real causado por filtros gruesos o apilados, lentes secundarias o un parasol de lente incorrecto.

- **Depuración de efectos de postprocesamiento.** Muestra los datos en tiempo real sobre los efectos de postprocesamiento. Estos datos se utilizan para depurar los efectos de postprocesamiento y poder ver los resultados de ajustar los efectos aplicados en tiempo real.

Capítulo 3

Renderizados de Unity 3D

3.1. Introducción al renderizado

En cada fotograma, el motor gráfico Unity efectúa la representación visual del entorno tridimensional en la pantalla del usuario mediante un procedimiento conocido como renderizado. Este proceso se ejecuta de acuerdo con una secuencia de operaciones organizada, denominada canalización de renderizado.

La ejecución del renderizado implica la recopilación exhaustiva de toda la información disponible a la escena tridimensional. Posteriormente, a través de una serie de pasos predefinidos, se realiza la transformación de dicha información en una imagen bidimensional, con el objetivo de ser visualizada en la pantalla.

Mientras el bucle de juego rige las dinámicas y sucesos dentro de la escena virtual, la canalización de renderizado regula la representación visual de dicha escena. Ambos elementos trabajan de manera conjunta para visualizar el entorno virtual creado en Unity.

Unity aporta tres tipos de renderizado:

3.2. Renderizado Estándar o Built-in

3.2.1. Introducción al renderizado Estándar

Aunque el renderizado Estándar es un punto de partida sólido y útil para familiarizarse con la creación de sombreadores, está progresivamente quedando en desuso.

En la actualidad, se alienta a los desarrolladores de juegos de Unity a utilizar los renderizados a través de scripts (SRP). Estos son el renderizado Universal (URP) y el renderizado de Alta Definición (HDRP), sobre el que se va a basar el presente proyecto. Estos dos últimos renderizados, con sus gráficos optimizados, herramientas de sombreado sin necesidad de programación, excelentes efectos de postprocesamiento y capacidades de soporte multiplataforma, se encuentran actualmente en el centro de atención, eclipsando al renderizado Built-in.

El renderizado Estándar presenta varias ventajas. La ventaja más evidente es su disponibilidad inmediata al crear un nuevo proyecto en Unity con una plantilla básica. Esto te permite familiarizarte con todo el conjunto de herramientas de shaders, materiales y renderizadores de malla. Dado que ha estado disponible durante mucho tiempo, el renderizado Estándar es compatible con una amplia variedad de sombreadores preexistentes, efectos visuales (VFX), herramientas de terceros y recursos de aprendizaje. [30]

A causa de que es renderizado de propósito general, aborda la mayoría de los casos de uso comunes que se pueden encontrar como desarrollador en Unity y es uno de los renderizados más ampliamente respaldados. Ya sea para un juego 2D o 3D, con iluminación en tiempo real o precalculada, con sondas de reflexión o iluminación global, con sistemas de partículas o shaders personalizados, con o sin postprocesamiento, el renderizado Estándar lo abarca todo. Con el tiempo adecuado y conocimiento, incluso puede generar representaciones bastante sólidas, como el de la Figura 3.1 correspondiente al proyecto de demostración "The Countyyard". [30]

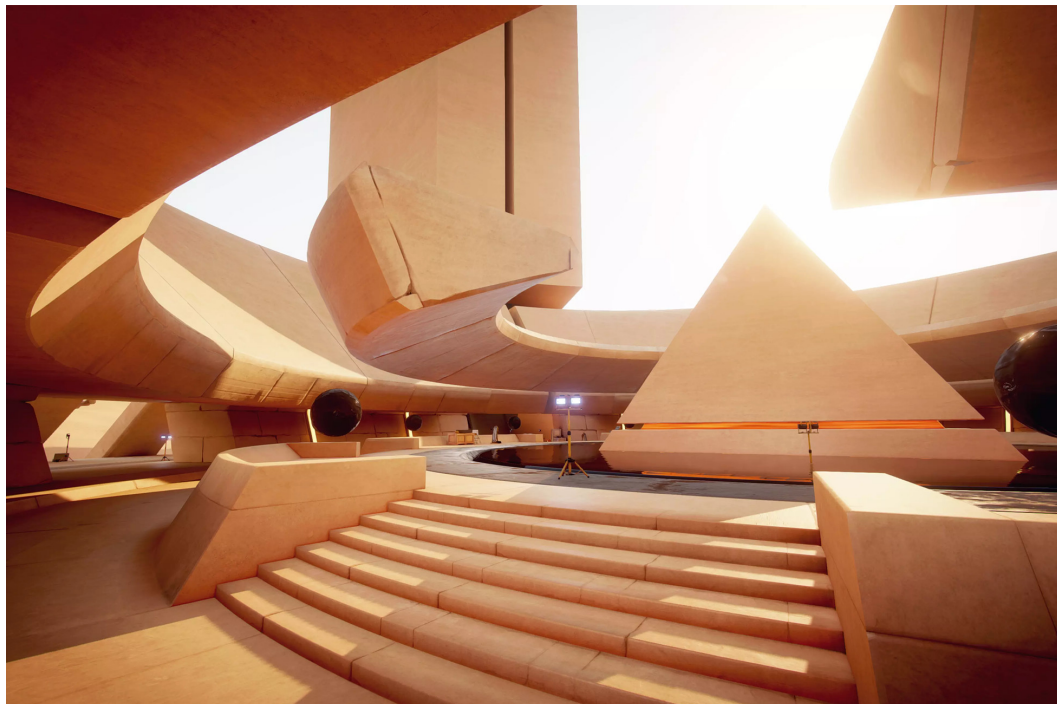


Figura 3.1: Proyecto Countyyard. Ejemplo con renderizado Estándar. Fuente: Tienda de activos de Unity.

La cuestión central radica en que la mayoría de estas funcionalidades están sujetas a limitaciones significativas. Aunque permite crear sistemas básicos, se encuentra rápidamente limitado en términos de rendimiento y no puede manejar más que un número limitado de partículas. Los módulos de postprocesamiento deben ser instalados mediante el paquete de Postprocesamiento V2. Por supuesto, la creación de shaders personalizados requiere conocimiento del lenguaje HLSL y la disposición para adentrarse en la programación de scripts. El renderizado Built-in compensa estos problemas al permitir una personalización exhaustiva y optimizaciones de fácil configuración.

No obstante, la característica interesante de este renderizado para los desarrolladores avanzados de sombreadores es su capacidad para intervenir en el proceso de renderizado y a los búfers de comandos.

La idea de los búfers de comandos es otorgar la capacidad de influir en el proceso de renderizado de la escena al permitir acumular uno o múltiples comandos de renderizado en un búfer y posteriormente ejecutar esta secuencia de comandos en un punto específico. Estos comandos pueden abarcar una amplia variedad de acciones, como renderizar una malla determinada, establecer una propiedad global de un shader o seleccionar el destino de renderización activo. Por lo tanto, estos búferes representan una herramienta interesante para extender el renderizado Estándar, e incluso para emular funcionalidades de renderizado presentes en otros renderizados. [31]

3.2.2. Técnicas de renderizado en el renderizado Estándar

El renderizado Built-in de Unity admite varias técnicas de renderizado. Una técnica de renderizado consiste en una secuencia de operaciones relacionadas con la iluminación y el sombreado. Cada una de ellas tiene sus propias capacidades y características de rendimiento. [32]

- **Renderizado directo**

El renderizado directo o Forward Rendering es la vía de renderizado predefinida en el renderizado Estándar de Unity y se caracteriza por ser una técnica de renderizado de propósito general.

El renderizado directo ejecuta el renderizado de cada objeto en una o varias pasadas, dependiendo de las luces que inciden sobre el objeto. Estas luces se manejan de manera distinta en el Renderizado directo, según su configuración e intensidad.

En el renderizado directo, un cierto número de las luces más brillantes que afectan a cada objeto dentro de la escena, se renderizan completamente en modo iluminado por píxel. A continuación, se calculan hasta cuatro luces pun-

tuales por vértice. El resto de luces se calculan mediante Armónicos Esféricos (SH), lo cual es mucho más rápido, pero es únicamente una aproximación.

Para la representación de cada objeto se realiza en primer lugar la Pasada Base (Base Pass), que aplica una luz direccional por píxel y todas las luces por vértice y SH. Las demás luces por píxel se representan en pasadas adicionales, de forma que se aplique una pasada por cada luz. [33]

■ Sombreado diferido

El sombreado diferido o Deferred Shading es la vía de renderizado que ofrece la máxima fidelidad en iluminación y sombras dentro del renderizado Estándar.

Este tipo de renderizado tiene la ventaja de que el procesamiento de la iluminación tiene una sobrecarga proporcional al número de píxeles que la luz ilumina. Esto se determina por el tamaño del volumen de luz en la escena, independientemente de cuántos objetos de juego ilumine. El sombreado diferido también tiene un comportamiento altamente consistente y predecible. El efecto de cada luz se calcula por píxel, por lo que no hay cálculos de iluminación tan gruesos.

Sin embargo, el sombreado diferido no tiene un soporte real para el antialiasing y no puede manejar objetos de juego semitransparentes. En este caso, para la representación de cada objeto se aplican los pases del búfer G y de iluminación. El búfer G es el encargado de representar cada objeto una vez. También se encarga de otras tareas como reflexiones, mapas de luz o colores difusos. El pase de iluminación calcula la iluminación en función del búfer G y la profundidad.

Los objetos con sombreadores que no admiten sombreado diferido se representan después de que se complete el sombreado diferido, utilizando el renderizado directo. [34]

■ Renderizado por vértices heredado

El renderizado por vértices heredado o Vertex Lit Rendering es la vía de renderizado con la fidelidad más baja en iluminación y sin soporte para sombras en tiempo real. Básicamente, se trata de un subconjunto de la vía de renderizado directo.

Generalmente representa cada objeto en un solo paso, calculando la iluminación de todas las luces para cada vértice. Dado que toda la iluminación se calcula a nivel de vértices, esta ruta de representación no admite la mayoría de los efectos por píxel como sombras, texturas proyectadas y reflejos especulares detallados. [35]

3.2.3. Orden de renderizado en el renderizado Estándar

En el renderizado Estándar, el orden en el que Unity representa objetos se basa en dos elementos: en qué cola de representación se encuentra el objeto y cómo Unity ordena los objetos dentro de esa cola de representación.

Unity clasifica los objetos en grupos llamados colas de representación y después los representa de uno en uno [36]. Existen las siguientes colas de representación, que se representan en el siguiente orden:

- **Fondo.** Este grupo se destina a elementos que requieren representación en la parte posterior de la escena.
- **Geometría.** En esta categoría se incluye la representación de objetos opacos.
- **Prueba alfa.** Esta cola se utiliza para representar geometría que requiere pruebas alfa y se ubica justo después de la cola de Geometría ya que resulta más eficiente representar objetos con pruebas alfa una vez que se han representado todos los objetos sólidos.
- **Transparente.** En este grupo se engloba cualquier elemento que implica mezcla alpha como objetos de vidrio o efectos de partículas.
- **Superposición.** Aquí se representan efectos que se muestran sobre todos los demás elementos, como los destellos de las lentes.

3.3. Renderizado Universal

3.3.1. Introducción al renderizado Universal

El renderizado Universal o URP constituye un sistema de procesamiento ligero diseñado con la finalidad de ofrecer una calidad gráfica sustancial, a la par de mantener un rendimiento óptimo. Este sistema respalda las técnicas de renderizado más recientes, abarcando la iluminación dinámica, la instanciación de GPU y los efectos de post-procesamiento. URP proporciona tres vías de renderizado: directo, directo+ y diferido. El renderizado Universal se manifiesta como un renderizado a través de scripts (SRP) desarrollado por Unity.

La concepción de URP por parte de Unity se orienta hacia la compatibilidad con una diversidad de plataformas, lo que implica su funcionalidad en una amplia variedad de dispositivos. Independientemente de si se está desarrollando un juego destinado a dispositivos móviles, o a ordenadores y consolas de gama alta, el renderizado Universal demuestra su eficacia.

El renderizado Universal está concebido con la intención de proveer a los desarrolladores de renderizado ligero optimizado para una extensa gama de plataformas, abarcando dispositivos móviles, consolas y ordenadores. Este pipeline adopta un enfoque de renderizado adelante, caracterizado por su optimización en términos de rendimiento y eficiencia, lo que facilita a los desarrolladores para generar gráficos

de alta calidad con requisitos de hardware mínimos. Asimismo, URP incorpora un modelo de sombreado simplificado que engloba capacidades básicas de iluminación y sombras, junto con el respaldo para efectos de post-procesamiento y una variedad de características adicionales. En términos generales, URP se configura para facilitar la publicación de juegos en hardware diverso. [37]

3.3.2. Características del renderizado Universal

El renderizado Universal de Unity ha surgido como una herramienta crucial en el desarrollo de juegos, ofreciendo una solución versátil con un amplio alcance y numerosos beneficios en comparación al renderizado Estándar. A continuación se muestran algunas de las características que contiene este tipo de renderizado. [38]

Eficiencia Ligera y centrada en el Rendimiento

La optimización para el renderizado Universal ha sido meticulosamente estructurada para proporcionar experiencias de juego fluidas, incluso en dispositivos de capacidad computacional más limitada. Su arquitectura de datos orientada minimiza la carga en la unidad central de procesamiento (CPU) y maximiza la eficiencia en la gestión de recursos computacionales.

Extensibilidad

La adaptabilidad inherente del renderizado Universal permite a los desarrolladores expandir su funcionalidad para satisfacer requisitos específicos del proyecto. La utilización de activos del renderizado a través de scripts otorga a los desarrolladores la capacidad de concebir características de renderizado personalizadas o modificar las existentes, ampliando así las capacidades del sistema.

Integración con el gráfico de sombreadores o Shader Graph

El renderizado Universal se fusiona de manera sinérgica con el gráfico de sombreado de Unity, posibilitando la creación de shaders visualmente impactantes y adaptables por parte de artistas y desarrolladores, prescindiendo de la necesidad de incurrir en complejidades de programación.

Compatibilidad multiplataforma

La uniformidad de rendimiento de URP en diversas plataformas conlleva una simplificación inherente en el desarrollo de proyectos que abarcan múltiples entornos. Los desarrolladores pueden concebir una única implementación que se despliegue en diversos dispositivos, mitigando así la complejidad y el esfuerzo asociados con la adaptación a diferentes plataformas.

Técnica de renderizado

El URP renderiza escenas utilizando la técnica del renderizado Directo, los modelos de sombreado para shaders incluidos con el renderizado Universal, la cámara y el activo UniversalRP que es incorporado al usar este renderizado. En el caso del renderizado directo, el renderizado Universal implementa un bucle de renderizado que indica a Unity cómo renderizar un fotograma.

El renderizado URP ejecuta un bucle de cámara para cada cámara, el cual realiza los siguientes pasos.

- Primero oculta objetos renderizados en tu escena.
- A continuación, construye datos para el renderizador.
- Finalmente, ejecuta un renderizador que produce una imagen en el búfer del fotograma.

Iluminación global y sombreado avanzado

El renderizado Universal ofrece capacidades avanzadas de iluminación global y sombreado, permitiendo la creación de entornos visualmente impactantes. Esto es esencial para juegos que buscan sumergir a los jugadores en mundos detallados y envolventes.

Postprocesamiento

El renderizado Universal incorpora su propia implementación de postprocesamiento, por lo que no es necesario instalar ningún otro paquete adicional. Cabe destacar que URP no es compatible con la versión 2 del paquete de postprocesamiento que se utiliza en el renderizado Estándar. Esta implementación utiliza el mismo sistema de Volumen que el renderizado de Alta Definición.

3.3.3. Ventajas de la Implementación del renderizado Universal

Accesibilidad

El renderizado Universal es accesible tanto para desarrolladores principiantes como para expertos. La interfaz intuitiva y las herramientas visuales, como Shader Graph, facilitan la creación de efectos visuales complejos sin requerir un profundo conocimiento técnico.

Flujo de trabajo sencillo

La incorporación del Gráfico de sombreadores en el renderizado Universal libera el proceso creativo y además descentraliza la dependencia de habilidades de programación, permitiendo expresar la creatividad de manera más directa. La programación visual facilita iteraciones rápidas y la experimentación con distintos estilos visuales, optimizando la eficacia del flujo de trabajo.

Optimización del rendimiento en dispositivos móviles

La concepción ligera del renderizado Universal adquiere una relevancia particular en el desarrollo de juegos móviles, asegurando no solo un rendimiento fluido, sino también una administración eficiente de recursos. Los dispositivos móviles pueden capitalizar el renderizado optimizado del renderizado Universal sin sacrificar la calidad visual, lo que resulta esencial para garantizar una experiencia de juego satisfactoria en este contexto.

Proyección futura del proyecto

En consonancia con la evolución continua de Unity, el renderizado Universal persiste como una solución de renderizado prospectiva, garantizando que los proyectos permanezcan pertinentes y optimizados ante futuros lanzamientos de la plataforma.

Enfoque unificado de renderizado

El renderizado Universal ejerce un enfoque uniforme en el proceso de renderizado, independientemente de la plataforma, simplificando el desarrollo y reduciendo la necesidad de realizar optimizaciones específicas para cada entorno de ejecución.

En resumen, el renderizado Universal emerge como un componente esencial en la caja de herramientas de los desarrolladores, permitiendo la creación de juegos de alto rendimiento y visualmente impactantes en diversas plataformas. Su diseño liviano, su capacidad de extensibilidad y su integración con Shader Graph lo convierten en una herramienta valiosa tanto para proyectos independientes de menor envergadura como para producciones de mayor escala. Al comprender las características y beneficios de este renderizado y al incorporarlo adecuadamente en el flujo de trabajo, los desarrolladores pueden elevar el estándar de sus proyectos de desarrollo de juegos, ofreciendo experiencias atractivas a los jugadores en cualquier dispositivo. La adopción de URP representa un paso hacia el futuro del renderizado en Unity, desbloqueando el potencial completo de los proyectos desarrollados en esta plataforma. [39]

3.4. Renderizado de Alta Definición

3.4.1. Introducción al renderizado de Alta Definición

El renderizado de Alta Definición o HDRP es una tecnología de renderizado avanzada de Unity. Este permite una calidad de imagen realista e impresionante. Este renderizado en Alta Definición está diseñado específicamente para superar los límites del renderizado 3D en tiempo real y lograr impresionantes gráficos de alta gama.

Un aspecto clave es calcular técnicas de iluminación avanzadas y efectos visuales en tiempo real. Esto permite utilizar cálculos de luz complejos y globales, iluminación volumétrica y reflejos en tiempo real, haciendo posible crear efectos de luces y sombras realistas.

Sin embargo, cabe señalar que este renderizado tiene requisitos de hardware más altos que el renderizado Estándar. Por lo tanto, el canal de renderizado de alta definición solo debe usarse en proyectos con hardware potente o para casos de uso donde la calidad visual es de gran importancia.

3.4.2. Características propias del renderizado en Alta Definición

Este renderizado fue diseñado específicamente para proyectos que tienen altas exigencias en cuanto a calidad de gráficos. Estos incluyen, entre otros, juegos, visualizaciones arquitectónicas y efectos cinematográficos. Para aprovechar todo su potencial, es importante comprender sus características más importantes. [40]

Sistema de materiales PBR

El sistema de materiales en el renderizado de Alta Definición es una parte integral y permite especificar las propiedades visuales de las superficies. El objetivo es crear materiales fotorrealistas basados en propiedades físicas que se muestren correctamente en diferentes condiciones de iluminación. Para ello, se utiliza un sistema de material de renderizado basado en la física (PBR), del inglés Physically Based Rendering. En Unity, se pueden tener en cuenta varias propiedades para un material PBR:

- **Mapa base (Albedo, color).** El mapa base representa el color base del material e influye en cómo se refleja y absorbe la luz. Aquí también se pueden utilizar texturas.
- **Normales.** Las normales se utilizan para determinar los detalles de la superficie de un modelo y permitir la simulación de reflejos de luz.
- **Rugosidad.** La rugosidad afecta qué tan lisa o rugosa es la superficie. Una superficie rugosa dispersa más la luz, mientras que una superficie lisa la refleja más.
- **Metálico.** La metalicidad define si un material es metálico o no metálico. Los materiales metálicos reflejan fuertemente la luz, mientras que los materiales no metálicos la dispersan de manera más difusa.

Se pueden utilizar texturas y mapas de detalles para obtener resultados aún más realistas. Las texturas contienen información de color que define con precisión el color de la superficie del material. Por ejemplo, se puede asignar una textura específica a la superficie.

Al utilizar texturas en el renderizado de Alta Definición, es importante encontrar el equilibrio adecuado entre calidad visual y rendimiento. Este renderizado admite varios métodos de compresión pudiendo reducir el tamaño de la textura sin reducir significativamente la calidad visual.

Efectos de iluminación

El renderizado de Alta Definición ofrece una amplia gama de técnicas de iluminación. Se pueden crear efectos de iluminación realistas que se pueden aplicar a modelos y escenas 3D en tiempo real.

■ Iluminación Global

Un aspecto central de la iluminación es la iluminación global. En ella, se calcula la propagación de la luz en toda la escena y también permite una distribución de la luz realista, reflejos e iluminación indirecta.

- **Iluminado (Enlighten)**. Es una solución probada y optimizada para aplicaciones en tiempo real y proporciona efectos de iluminación globales precisos. El enlighten ofrece un buen equilibrio entre calidad y rendimiento para la mayoría de las escenas.
- **Mapeado de luz progresivo (Progressive Lightmapper)**. Este algoritmo utiliza la técnica de hornear e iterar para lograr gradualmente una iluminación global de alta calidad. Es particularmente útil para escenas estáticas porque precalcula la iluminación y realiza mejoras iterativas.
- **Trazado de rayos (Ray Tracing)**. Con el soporte de Ray Tracing, se puede lograr una iluminación global altamente realista. Permite una representación aún más realista de reflejos, sombras e iluminación indirecta.

■ Luces dinámicas y sombras en tiempo real

El renderizado de Alta Definición admite fuentes de luz dinámicas que pueden moverse e iluminar la escena en tiempo real. Estos incluyen focos, luces puntuales, luces direccionales y otras fuentes de luz.

Sin embargo, el renderizado de Alta Definición admite una mayor cantidad de fuentes de luz dinámicas en tiempo real. Además, las fuentes de luz se ajustan en función de propiedades físicas como la temperatura de la luz en grados Kelvin y la intensidad de la luz en lúmenes, candelas o lúxes.

Las fuentes de luz dinámicas también permiten la creación de sombras dinámicas que cambian según la posición y el movimiento de la fuente de luz en la escena. Al combinar fuentes de luz dinámicas con sombras en tiempo real, la escena parece muy realista.

■ Iluminación volumétrica

Otra funcionalidad es el soporte para iluminación volumétrica. Esto permite visualizar los rayos de luz que se encuentran dispersos en la atmósfera. Así cuando interactúan con partículas, niebla, humo u otros objetos de volumen.

Esto puede crear efectos como rayos de luz (godrays), niebla volumétrica y refracciones de luz en la atmósfera. Los rayos de luz añaden un efecto atmosférico a la escena y son particularmente efectivos en escenas con iluminación dinámica.

La niebla volumétrica es otra aplicación de la iluminación volumétrica. Esto crea una representación realista de la niebla que se extiende por una escena. La nebulosa está iluminada por rayos de luz, lo que permite representar luces y sombras en una atmósfera densa. Esto también se puede ver en la Figura 3.2.



Figura 3.2: Niebla volumétrica en el renderizado de Alta Definición. Fuente: Unity.

La iluminación volumétrica también permite visualizar las refracciones de la luz. Esto ocurre cuando la luz atraviesa un objeto sólido transparente como agua o vidrio. Los rayos de luz cambian de dirección a medida que atraviesan el objeto sólido, creando impresionantes efectos visuales. Esto es particularmente útil para escenas que involucran cuerpos de agua, estructuras de vidrio u otros materiales transparentes.

Postprocesamiento y efectos visuales

El postprocesamiento es un paso importante en el proceso de renderizado que permite ajustar el aspecto de una escena después de la fase de renderizado real. El uso de efectos de postprocesamiento puede mejorar la apariencia visual de la escena y permite agregar efectos especiales.

Una variedad de efectos de postprocesamiento (véase 2.3.4) están disponibles en el renderizado de Alta Definición. Estos efectos se aplican a la imagen renderizada después de renderizar la escena. Esto permite ajustar la imagen final con filtros especiales y otras manipulaciones.

El renderizado de Alta Definición ofrece efectos visuales avanzados diseñados específicamente para gráficos fotorrealistas y visualizaciones gráficas de alta calidad. Algunos de los efectos visuales avanzados son:

- **Desenfoco de movimiento.** Este efecto simula el desenfoco que se produce cuando los objetos o la cámara se mueven rápidamente. Esto ayuda a que

los movimientos en la escena sean realistas y le da a la imagen un dinamismo adicional.

- **Reflejos del espacio de pantalla (SSR).** Los reflejos del espacio de pantalla son un método para mostrar reflejos que se reflejan en tiempo real en las superficies del entorno. Esto permite crear reflejos realistas en superficies lisas.
- **Oclusión ambiental.** Simula las sombras creadas en espacios pequeños y confinados. Este efecto crea sombras realistas y resalta los detalles de la escena.

Aunque los efectos de postprocesamiento producen resultados visualmente atractivos, la aplicación de estos efectos puede afectar el rendimiento. Para equilibrar estos altos requisitos y permitir que los proyectos con el renderizado de Alta Definición se ejecuten en máquinas menos potentes, este renderizado tiene algunos trucos adicionales.

Gráficos de sombreadores o Shader Graphs

Para hablar del renderizado de Alta Definición, es esencial destacar que este hace uso de sombreadores y unidades de iluminación distintas a las empleadas en el renderizado Estándar. Los shaders en este tipo de renderizado se pueden crear tanto a través de HLSL como utilizando el Shader Graph, una nueva herramienta añadida para este tipo de renderizado. Esta herramienta permite a los creadores no técnicos crear efectos visuales sin necesidad de tener conocimiento de programación en dicho campo. [41]

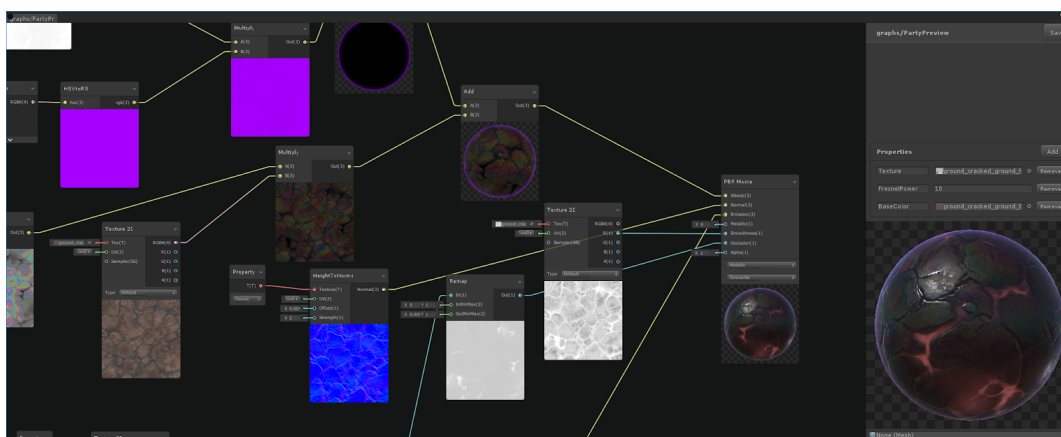


Figura 3.3: Grafo de composición de un Gráfico de sombreador basado en nodos. Fuente: Forum Unity.

El Gráfico de sombreadores es una función que proporciona un editor visual para crear sombreadores complejos. Utiliza nodos que están conectados entre sí para crear una representación similar a un diagrama de flujo de la funcionalidad del sombreador

como muestra la Figura 3.3. Es un método intuitivo para crear sombreadores, ya que le permite visualizar los componentes y el flujo del código del sombreador.

Shader Graph proporciona una forma mucho más sencilla de crear sombreadores en comparación con la codificación de sombreadores tradicional. En lugar de tener que escribir líneas de código complejas, permite visualizar las operaciones en un gráfico basado en nodos. Esto proporciona retroalimentación instantánea y permite un desarrollo de sombreadores mucho más rápido y eficiente. [42]

Antialiasing

El aliasing es un efecto secundario que ocurre cuando un muestreador digital toma muestras de información del mundo real e intenta digitalizarla. Por ejemplo, cuando muestreas audio o vídeo, el aliasing significa que la forma de la señal digital no coincide con la forma de la señal original.

Esto es más obvio si se compara las señales originales y digitales de una fuente visual en sus detalles más pequeños. El procesamiento de señales regular utiliza la velocidad de Nyquist para evitar el aliasing. Sin embargo, no es práctico para la representación de imágenes porque requiere muchos recursos. [43]

Para evitar bordes dentados y aliasing no realista, el renderizado de Alta Definición nos proporciona herramientas de antialiasing tanto a nivel de soporte físico como a nivel de programas. Es posible utilizar el antialiasing multimuestra (MSAA) para mejorar drásticamente los efectos visuales, aunque a un costo computacional muy alto.

Por otro lado, este renderizado permite utilizar otros tipos de antialiasing que se aplican como efectos de postprocesamiento y reducen el aliasing para un menor consumo computacional, como el antialiasing de aproximación rápida (FXAA) o el antialiasing temporal (TAA).



Figura 3.4: Comparación entre valores de recuento de muestras MSAA. Fuente: The danger zone.

Cuando se utiliza el MSAA, el recuento de muestras MSAA determina cuántas muestras por píxel utiliza el efecto y, por lo tanto, cuán suaves serán los bordes y cuán exigente será el antialiasing. La Figura 3.4 muestra la diferencia entre una representación escalada sin aliasing y con el recuento de muestras MSAA configurado en 2X, 4X y 8X. [44]

3.4.3. Beneficios del renderizado de Alta Definición

El renderizado de Alta Definición es una herramienta poderosa para lograr una calidad gráfica superior. Algunos de los beneficios clave se explican a continuación [40].

- **Calidad de gráficos fotorrealistas.** El renderizado de Alta Definición permite la creación de gráficos fotorrealistas en tiempo real. Gracias a técnicas de iluminación avanzadas y materiales de alta calidad, se pueden desarrollar escenas con sombras, reflejos y efectos de iluminación realistas.
- **Representación basada en la física (PBR).** Al utilizar el sistema de materiales basados en la física, se pueden representar propiedades físicas reales como el albedo, es decir, la capacidad que tiene cualquier cuerpo de reflejar una radiación incidente, el brillo y la rugosidad. Esto permite que los materiales se muestren de manera realista, independientemente de las condiciones de iluminación y el ángulo de visión.
- **Técnicas de iluminación avanzadas.** El proceso de renderizado de Alta Definición se puede utilizar para crear escenarios de iluminación complejos.

Esto incluye sombras realistas, iluminación global, reflejos en tiempo real y efectos volumétricos.

- **Efectos de postprocesamiento.** La tecnología de renderizado integra efectos de postprocesamiento para mejorar la apariencia visual después de la fase de renderizado real. Se pueden utilizar efectos como floración, mapeo de tonos, profundidad de campo y corrección de color para ajustar el aspecto de una escena.

En general, el renderizado de Alta Definición ofrece una solución potente y fácil de usar que lleva la calidad visual de los proyectos a un nivel más elevado. Con una gran cantidad de técnicas de renderizado realistas y efectos visuales, es excelente para proyectos que requieren gráficos de alta calidad.

3.4.4. Técnicas de renderizado en Alta Definición

El renderizado de Alta Definición otorga también la capacidad de representar materiales iluminados utilizando tanto la técnica de renderizado directo como el sombreado diferido como ya se comentó en el primer apartado de este capítulo (véase 3.2.2).

La técnica de sombreado Diferido en el renderizado de Alta Definición es más veloz, particularmente en entornos que contienen una variedad de materiales y múltiples fuentes de luz locales. Sin embargo, si se incorpora una única luz direccional en la escena, puede beneficiarse de un mayor rendimiento.

La técnica de renderizado directo y el sombreado diferido implementan las mismas características, no obstante, la calidad puede presentar variaciones entre ambos. Esto implica que el renderizado de Alta Definición es compatible con todas las características, independientemente de la técnica de renderizado seleccionada. [45]

A continuación, se muestra una comparativa de las principales características de cada tipo de renderizado.

- **Desviación de sombras normal.** Para el renderizado directo se utiliza la normal geométrica (la normal de vértice) del material para la desviación de sombras, reduciendo así la aparición de artefactos de sombras. En el sombreado diferido, utiliza la normal de píxel del material para la desviación de sombras, lo que resulta en una mayor cantidad de artefactos de sombras.
- **Color Emisivo.** En el renderizado directo la oclusión ambiental no afecta al color emisivo mientras que en el sombreado diferido sí afecta debido a las limitaciones técnicas de este renderizado.

- **Oclusión Ambiental.** Para el renderizado directo, se aplica la cantidad mínima de oclusión ambiental. Esto da como resultado un oscurecimiento correcto. Sin embargo, en el sombreado diferido resulta un oscurecimiento incorrecto.
- **Calidad del Material.** En el sombreado diferido comprime las propiedades del material, como las normales o las tangentes, en el Búfer G, lo que da como resultado artefactos de compresión. Por otro lado, en el renderizado directo no hay compresión, por lo que no hay artefactos de compresión.
- **Prepaso de Profundidad.** Para el renderizado directo siempre renderiza un prepaso de profundidad, que genera una imagen de profundidad y un búfer de normales, mientras que en el diferido es opcional.

3.5. Comparativa entre los renderizados Universal y Alta Definición

Como ya se comentado previamente en este capítulo, una de las características clave de Unity es su sistema de renderizado, que permite a los desarrolladores crear gráficos de alta calidad y efectos visuales en tiempo real.

3.5.1. Análisis de características técnicas y diferencias

A continuación, se realiza una comparativa de las características más destacable de ambos renderizados basados en scripts y con calidad superior al renderizado Estándar. [46]

Cámara

Ambos soportan HDR y MSAA. El renderizado de Alta Definición utiliza propiedades físicas de la cámara mientras que el renderizado Universal no. El renderizado Universal está en investigación para la apilación, mientras que HDRP no lo soporta.

Luces en Tiempo Real

Ambos admiten luces direccionales, de punto y de área, pero el renderizado de Alta Definición tiene límites de luz más altos y permite sombras ilimitadas para la luz direccional principal. También admite luces SH, mientras el renderizado Universal no lo tiene desarrollado. En sombras en tiempo real, URP tiene opciones de cascada, mientras que HDRP tiene opciones más avanzadas.

Sombreado y eliminación

El renderizado Universal utiliza un sombreado de pase único, mientras que el renderizado de Alta Definición utiliza un enfoque agrupado. Ambos admiten eliminación por objeto y capa.

Procesamiento por lotes y espacio de color

Ambos admiten procesamiento por lotes estático y dinámico, así como instancias de GPU, pero HDRP no soporta el espacio de color Gamma.

Iluminación global

Ambos admiten iluminación global, pero el renderizado de Alta Definición tiene más opciones avanzadas como iluminado en tiempo real y GPU progresiva. Además, tiene un soporte más extenso para la mezcla de sondas de luz y volúmenes de proxy (LPPV).

Componentes de efectos visuales

Ambos carecen de soporte para Halo y destello de lente. HDRP requiere el uso de efectos gráficos para efectos de estela personalizados.

Sombreadores y materiales avanzados

Ambos admiten el Gráfico de sombreadores y presentan diferencias en la implementación de materiales avanzados como la capa de transparente. El renderizado de Alta Definición ofrece opciones avanzadas como dispersión subsuperficial e iridiscencia.

Postprocesamiento

Ambos admiten varias funciones de postprocesamiento, pero el renderizado de Alta Definición tiene más opciones específicas, como la exposición automática basada en la adaptación ocular.

Partículas y Terreno

Ambos admiten efectos gráficos y sistemas de partículas, pero HDRP tiene más opciones avanzadas para materiales y sombreado. Sin embargo, no admite algunas características específicas de terreno presentes en URP.

3.5.2. Rendimiento y fidelidad visual

Una de las principales diferencias entre los renderizados Universal y de Alta Definición es su nivel de rendimiento y fidelidad visual. URP está optimizado para el rendimiento y la eficiencia, lo que lo hace ideal para proyectos que requieren un alto rendimiento en una amplia gama de plataformas. Sin embargo, el renderizado de Alta Definición sacrifica parte de la fidelidad visual y capacidades avanzadas de iluminación y sombreado presentes en HDRP para lograr este nivel de rendimiento.

El renderizado de Alta Definición, por otro lado, está optimizado para la fidelidad visual y gráficos fotorrealistas, siendo ideal para proyectos que requieren gráficos de alta gama y efectos visuales avanzados. Sin embargo, requiere requisitos de hardware más potentes y puede no ser adecuado para dispositivos o plataformas de gama baja.

3.5.3. Complejidad y facilidad de uso

Otra diferencia clave entre URP y HDRP es su nivel de complejidad y facilidad de uso. El renderizado Universal está diseñado para ser fácil de usar y es relativamente simple de configurar, haciéndolo ideal para desarrolladores nuevos en Unity o que necesitan renderizado ligero. Sin embargo, el renderizado Universal puede resultar limitado para desarrolladores que requieren capacidades avanzadas de iluminación y sombreado.

El renderizado de Alta Definición, por otro lado, es más complejo y puede requerir más configuración, siendo ideal para desarrolladores experimentados que necesitan gráficos y efectos visuales avanzados. Sin embargo, puede resultar abrumador para desarrolladores nuevos en Unity o que necesitan un renderizado más simple.

3.5.4. Elección del renderizado para el presente trabajo

En resumen, los renderizados Universal y de Alta Definición son dos renderizados poderosos de Unity que ofrecen una amplia variedad de herramientas y características para crear gráficos y efectos visuales de alta calidad. URP está optimizado para el rendimiento y la eficiencia, siendo ideal para proyectos que requieren soporte en

diversas plataformas y bajos requisitos de hardware. HDRP, por otro lado, está optimizado para la fidelidad visual y gráficos fotorrealistas, siendo ideal para dispositivos de alta gama y proyectos que requieren capacidades avanzadas de iluminación y sombreado. [46]

Es por ello que se ha decidido optar por el renderizado de Alta Definición para la transformación de la visualización del gemelo digital del presente proyecto, debido a que este renderizado aporta una calidad visual superior y herramientas avanzadas. Este renderizado garantiza una representación más realista y establece una base sólida para futuras aplicaciones que requieren una precisión visual excepcional en simulaciones industriales de alta fidelidad.

Capítulo 4

Entorno del gemelo digital

4.1. Características de la visualización con el renderizado Estándar

El presente trabajo introduce una interesante investigación dentro del entorno Unity, utilizando el renderizado Estándar anteriormente introducido, como punto de partida. Este proyecto se centra en la mejora en la visualización de un gemelo digital. La esencia de este enfoque radica en la integración y evolución hacia el sistema de renderizado de Alta Definición de Unity, con el objetivo de ofrecer una calidad visual de mayor nivel y una fidelidad gráfica que refleje de manera precisa su representación del mundo real.

4.1.1. Componentes del gemelo digital

A continuación se analizan las partes que conforman el gemelo digital que va a ser objeto de estudio a lo largo del presente trabajo:

- **Nave.** La Figura 4.1 muestra una nave industrial. Es un edificio de uso industrial que pone la producción y/o almacena los bienes industriales, junto con la mano de obra, las máquinas que los generan, el transporte interno, la salida y entrada de mercancías, entre otras. En este caso es una estructura genérica.



Figura 4.1: Visualización de la nave en el proyecto. Fuente: Elaboración propia.

- Extrusora.** La extrusora de la Figura 4.2 es la encargada de la extrusión de polímeros mediante la acción del prensado, fusión, moldeado, presión y empuje de los materiales. El resultado es un molde nuevo que tendrá la forma deseada según la forma y diseño del tornillo de potencia (husillo) utilizado en la máquina.

La extrusión es un proceso continuo, en el cual los productos acabados se obtienen forzando material fundido a través de una herramienta de conformación (hilera, cabezal de extrusión, orificio).

- Mezcladora.** La mezcladora presente en la Figura 4.3 es una máquina destinada para la producción a gran escala cuyo propósito es utilizar un tanque de grandes proporciones para mezclar diversos tipos de materiales para la fabricación de producto

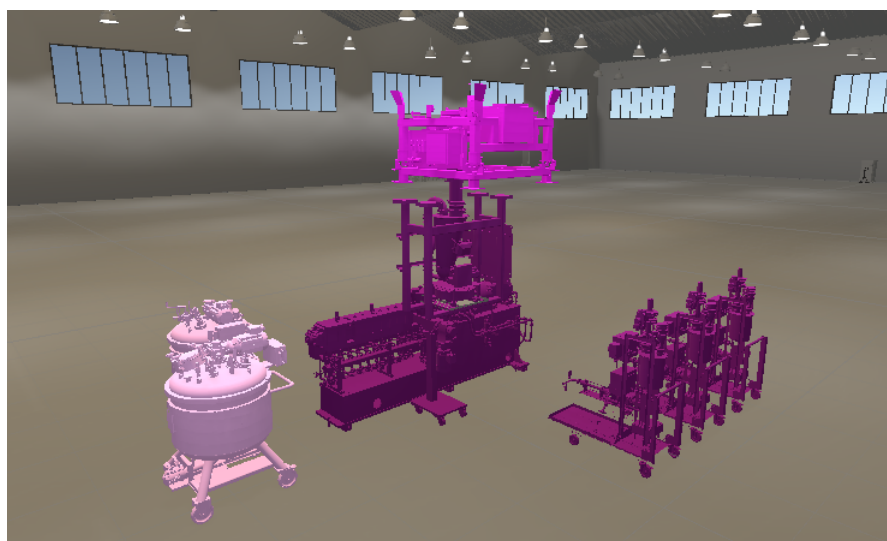


Figura 4.2: Visualización de la extrusora en el proyecto. Fuente: Elaboración propia.

s.

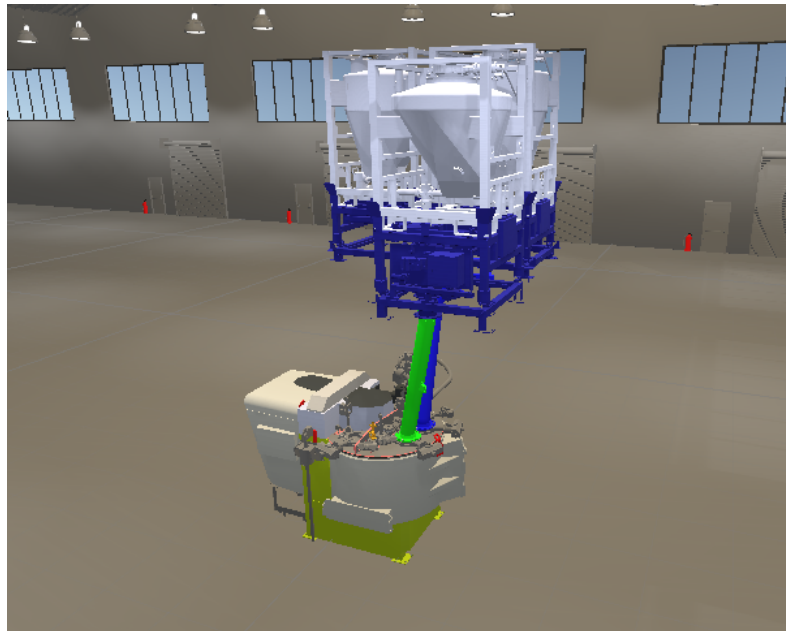


Figura 4.3: Visualización de la mezcladora en el proyecto. Fuente: Elaboración propia.

- **Armario de control.** La Figura 4.4 es una herramienta que ayuda a controlar los sistemas de energía a través de dispositivos de conexión que tienen la función de maniobrar, medir y resguardar la seguridad de toda una instalación, para que la misma funcione adecuadamente. Por consiguiente, es una pieza fundamental en toda ordenación eléctrica.
- **Sacos big bag.** Los sacos big bag, también conocidos como FIBC (Flexible Intermediate Bulk Containers), son envases para residuos que almacenan, mantienen y transportan tanto productos a granel como desechos para su eliminación. La Figura 4.5 muestra el saco presente en el proyecto.

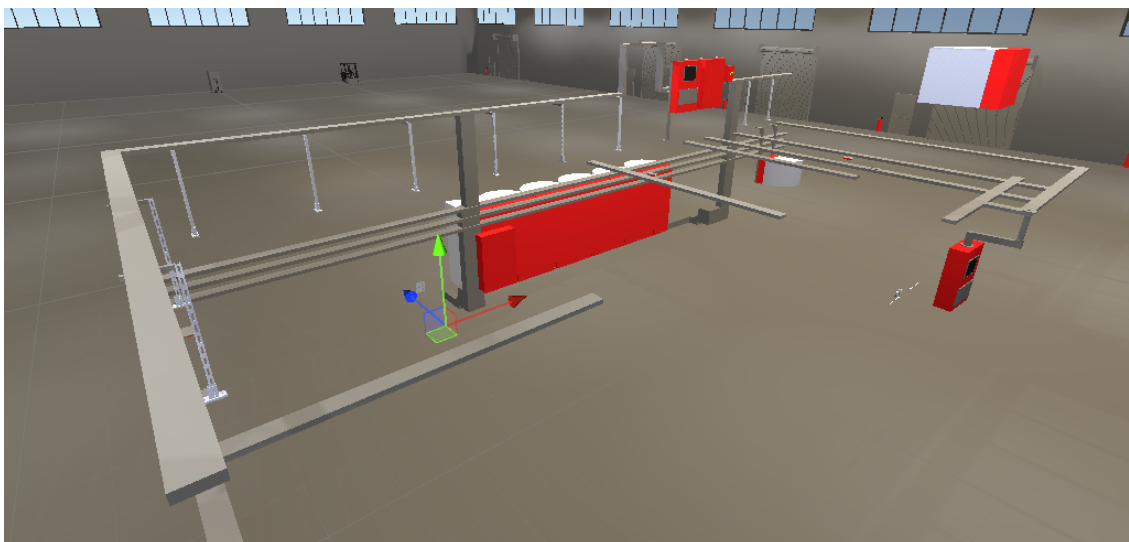


Figura 4.4: Visualización del armario de control en el proyecto. Fuente: Elaboración propia.

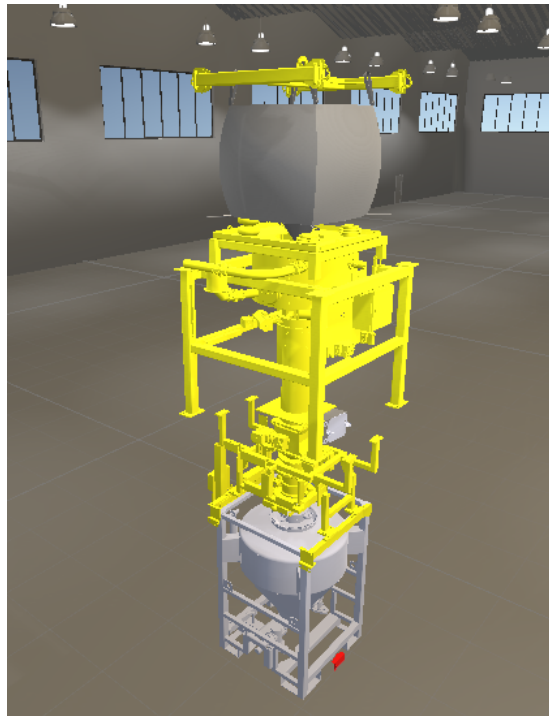


Figura 4.5: Visualización del saco big bag en el proyecto. Fuente: Elaboración propia.

- **Armario para bidones.** Estos bidones similares al de la Figura 4.6 son los recipientes idóneos para almacenar y trasladar todo tipo de líquidos o de materiales desmenuzados, es decir, en polvo, en almacenes, centros logísticos y fábricas.

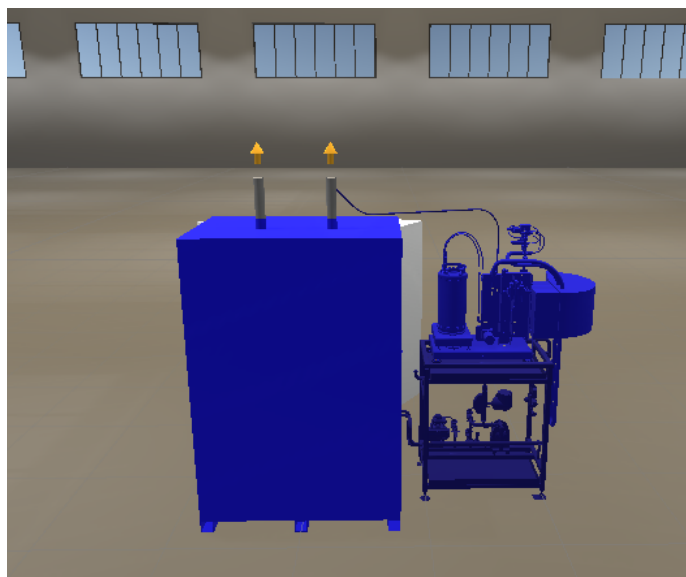


Figura 4.6: Visualización del armario para bidones en el proyecto. Fuente: Elaboración propia.

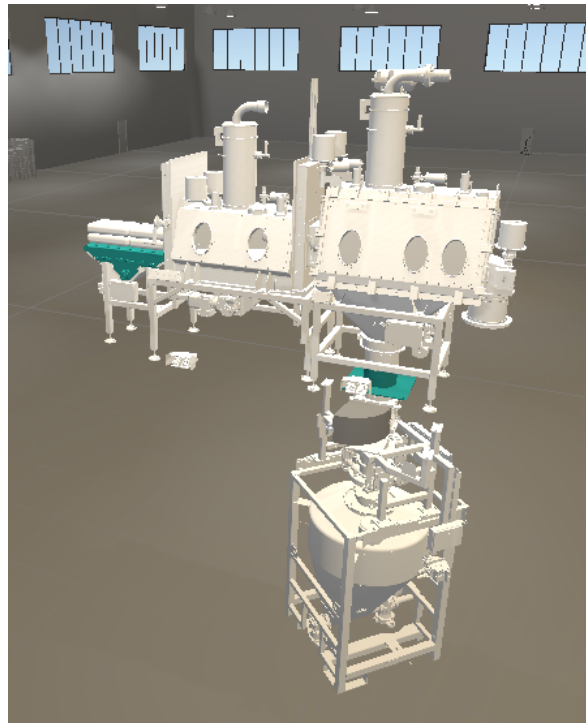


Figura 4.7: Visualización de la caja de guantes en el proyecto. Fuente: Elaboración propia.

- **Caja de guantes.** La caja de guantes que muestra la Figura 4.7 es una cabina o contenedor sellado que forma parte del equipamiento de laboratorio. Está diseñada para permitir una manipulación de objetos cuando se desea que estén separados del operador.
- **Grúa portátil.** Las grúas portátiles son un tipo de grúa que se puede transportar y mover fácilmente de un lugar a otro debido a su diseño compacto y peso ligero. El gemelo digital contiene una grúa como se observa en la Figura 4.8.

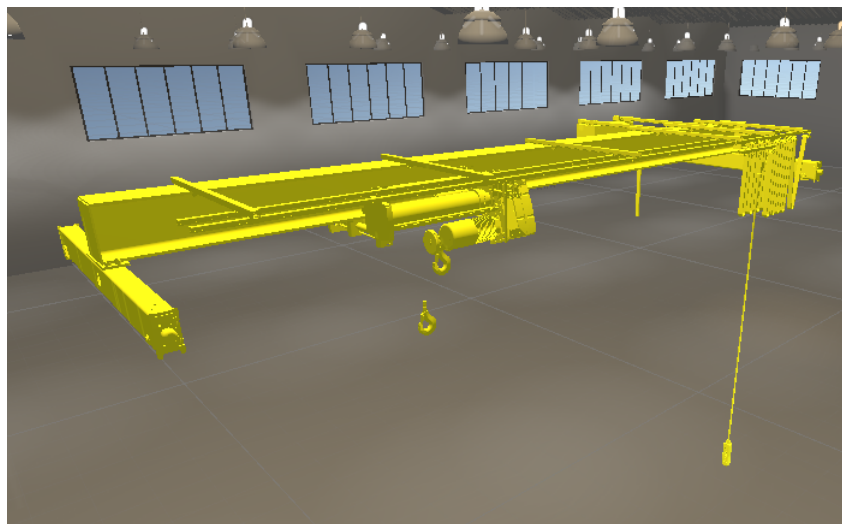


Figura 4.8: Visualización de la grúa en el proyecto. Fuente: Elaboración propia.

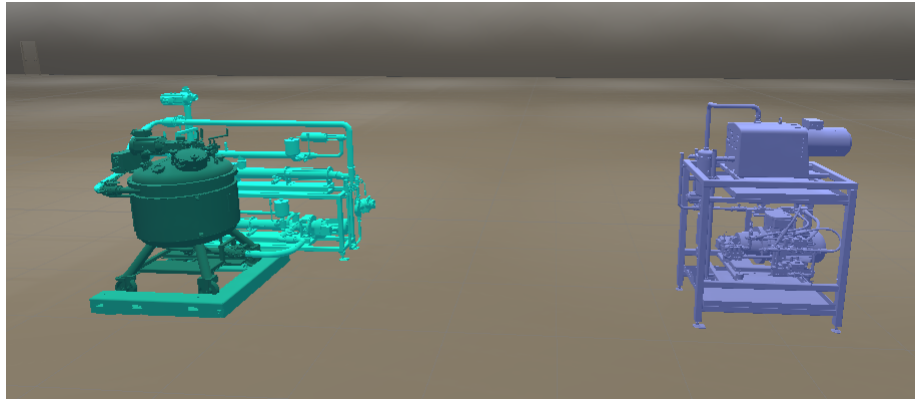


Figura 4.9: Visualización de la bomba hidráulica en el proyecto. Fuente: Elaboración propia.

- **Bomba hidráulica.** Una bomba hidráulica o bomba de agua como se ve en la Figura 4.9, es una máquina generadora que transforma la energía con la que es accionada en energía del fluido incompresible que mueve. El fluido incompresible puede ser líquido o una mezcla de líquidos y sólidos como puede ser el hormigón antes de fraguar o la pasta de papel.
- **Descargador de sacos.** Los descargadores de sacos que muestra la Figura 4.10, se utilizan para la descarga y vaciado de sacos o big bags. Se trata básicamente de un equipo en el que se vierte el contenido de los sacos por la parte superior en forma de rampa que cae por gravedad hacia una tolva y, dependiendo de la unidad, puede tener mecanismos de control de polvo o aspiración. Una vez retenido en la tolva, el material puede distribuirse según convenga.

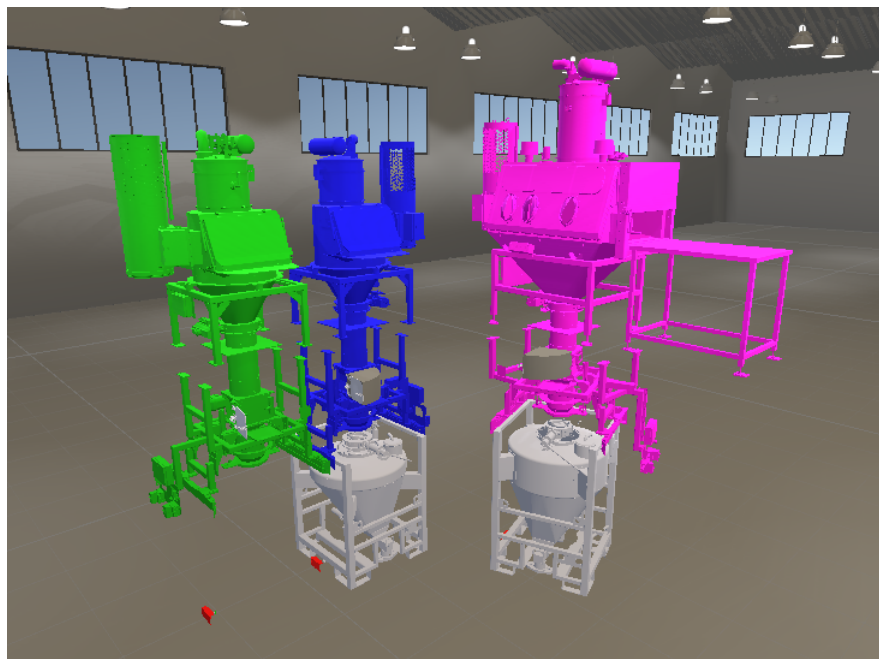


Figura 4.10: Visualización de los descargadores de sacos en el proyecto. Fuente: Elaboración propia.

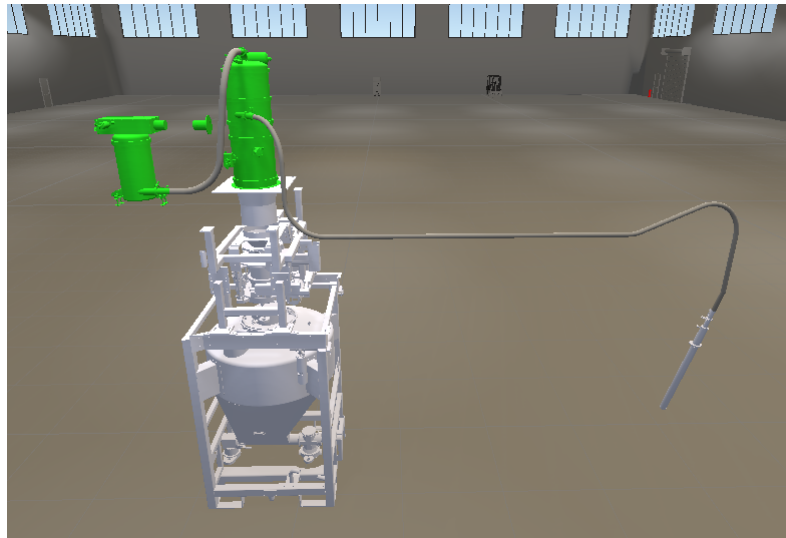


Figura 4.11: Visualización del transportador de vacío en el proyecto. Fuente: Elaboración propia.

- **Transportador de vacío.** Son sistemas para transportar polvos, granulados y otros tipos de materiales secos de un área de proceso a otra mediante vacío. Estos sistemas están compuestos de 4 elementos básicos: una fuente de vacío, un dispositivo de recepción del material, una línea de transporte y un filtro separador de aire y material. La Figura 4.11, muestra el transportador de vacío presente en el proyecto.
- **Estructura de acero.** Son el conjunto de elementos que forman la parte resistente de la construcción. El sistema estructural en acero se basa en un marco o esqueleto de cualquier estructura cuyos componentes en su mayoría son columnas y vigas de acero, que ayudan a soportar el resto de los elementos. La estructura de acero del gemelo digital se muestra en la Figura 4.12.

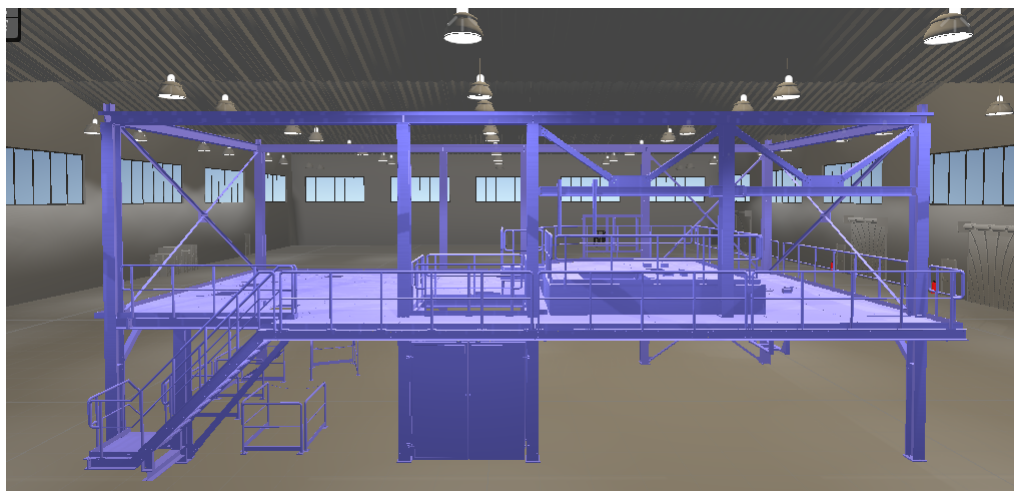


Figura 4.12: Visualización de la estructura de acero en el proyecto. Fuente: Elaboración propia.

Una vez explicados y mostrados todos los componentes que conforman la visualización del gemelo digital con el que se va a trabajar, la Figura 4.13 muestra el conjunto completo.

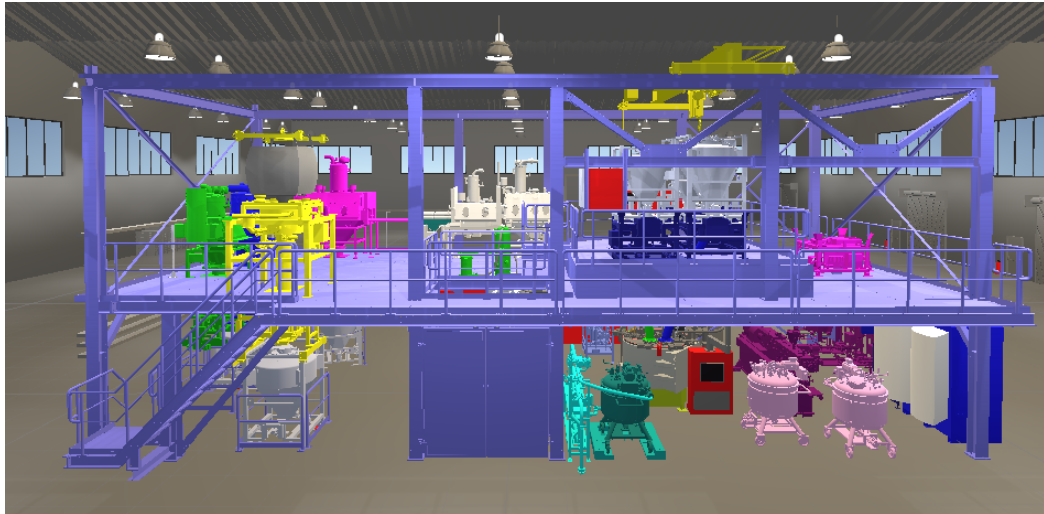


Figura 4.13: Visualización del gemelo digital al completo en el proyecto. Fuente: Elaboración propia.

4.1.2. Limitaciones en la visualización del gemelo digital

Como ya se ha explicado anteriormente y como además se puede observar en las imágenes previamente mostradas, a pesar de que el renderizado Estándar sea bastante versátil, presenta determinadas limitaciones en cuanto a la representación visual del gemelo digital.

Una de las limitaciones más evidentes al utilizar este renderizado para la representación del gemelo digital es la restricción en la fidelidad gráfica. Esto significa que la representación visual carece notablemente de detalles y efectos visuales avanzados que permiten una semejanza con el objeto real que se pretende simular.

Otro de los detalles a tener en cuenta es la representación lumínica dentro la escena donde se encuentra el gemelo digital, ya que los efectos que ofrece este renderizado en cuanto a sombras, reflejos y efectos de luz sobre los objetos tienen notables carencias. Esto también está relacionado con el efecto de postprocesamiento que soporta el renderizado Estándar. Como ya se comentó, este renderizado no aplica un efecto de postprocesamiento directo, sino que debe añadirse de forma externa. A pesar de introducir este grupo de efectos visuales, la imagen que ofrece no es fiel con la realidad a la que se pretende acercar.

Asimismo, la configuración que intenta ofrecer mejoras en cuanto a la visualización de los objetos contiene grandes limitaciones que no permiten mejorar signifi-

cativamente el entorno digital. Se ha intentado mejorar mediante la edición de los parámetros visuales que ofrece el paquete de postprocesamiento. Sin embargo, el intento no fue fructífero, ya que no aportó una mejora significativa en los detalles, como se buscaba. Es por esta razón por la que se ha descartado la inclusión de la misma, al no resultar de interés.

Tras el análisis de las limitaciones que se han comentado anteriormente, se ha decidido dar el paso a un renderizado de mayor definición con el fin de mejorar la calidad de la visualización de la réplica virtual que se quiere representar y que además sirva como base para futuros proyectos que quieran utilizar un renderizado más avanzado y de mayor calidad visual.

4.2. Conversión de la visualización del gemelo digital a Alta Definición

En el marco de este trabajo, se ha decidido transformar la imagen del gemelo digital presentado previamente en el entorno Unity, desde el renderizado Estándar hacia el renderizado de Alta Definición, al ser más avanzado y potente. Este cambio no es simplemente un ajuste técnico, sino una inmersión en el concepto de visualización del gemelo digital en Alta Definición, buscando no solo mejorar la estética visual sino también elevar la calidad general de la representación en tiempo real.

Antes de realizar esta transformación, se ha llevado cabo una evaluación exhaustiva de nuestro proyecto actual, analizando activos gráficos, materiales y configuraciones de luz utilizadas en el renderizado Built-in, de forma que se han podido identificar las áreas a mejorar. La compatibilidad y la optimización son consideraciones clave, ya que ciertos efectos o funciones pueden requerir ajustes manuales para funcionar de manera eficiente en el renderizado de Alta Definición.

El objetivo no es solamente completar la migración, sino además documentar y analizar los resultados obtenidos con el fin de aplicar estas mejoras en futuros proyectos.

4.2.1. Pasos realizados para aplicar la conversión

La conversión del proyecto del presente trabajo desde el renderizado Estándar hacia el renderizado de Alta Definición es un proceso detallado que implica ajustes significativos en diversos aspectos del entorno 3D. A continuación, se describe el enfoque adoptado.

- **Evaluación preliminar.** Se realiza un análisis de los activos gráficos, ma-

teriales y configuraciones de luz empleadas en el renderizado Estándar y se identifican las características específicas del proyecto que podrían beneficiarse de las capacidades mejoradas de HDRP, como la iluminación avanzada, las sombras precisas y los efectos visuales más realistas.

- **Preparación de activos.** La conversión empieza con la preparación de los activos que conforman el gemelo digital. Posteriormente, se actualizan los materiales y sombreadores para adaptar la compatibilidad con renderizado de Alta Definición, requiriendo tanto ajustes manuales como automatizados gracias al asistente que proporciona Unity para realizar dicha conversión. Algunos shaders personalizados requirieron modificaciones para alinearse con las nuevas características y funcionalidades de HDRP. Asimismo, se evaluó la compatibilidad de texturas y modelos, asegurándose de que estuvieran optimizados para el nuevo renderizado de Alta Definición.
- **Configuración de iluminación y cámaras.** Una parte fundamental de la adaptación al nuevo renderizado fue revisar y ajustar las configuraciones de iluminación y cámaras. Como ya se ha explicado, el renderizado de Alta Definición ofrece un conjunto diferente de opciones y parámetros, por lo que se configuraron las luces y la cámara existentes para aprovechar al máximo las características específicas de este renderizado en el apartado de iluminación. Esto incluye ajustar la intensidad de la luz, la temperatura del color y otros atributos para lograr una representación visual coherente y mejorada que represente fielmente la realidad.
- **Adaptación de efectos visuales.** La transformación también involucra la adaptación de efectos visuales. Efectos de postprocesamiento, como antialiasing, bloom y profundidad de campo, fueron revisados y ajustados para garantizar una apariencia visual coherente y atractiva en dentro el marco del renderizado de Alta Definición. La utilización de nuevas capacidades, como la renderización de profundidad y texturas, se consideró para mejorar la calidad general de la representación.

Esta conversión no es simplemente una transición técnica, sino un proceso que busca aprovechar las capacidades avanzadas que ofrece Unity con el renderizado de Alta Definición para mejorar la representación visual en el contexto de la creación de gemelos digitales más inmersivos y detallados.

4.3. Comparación de la visualización en ambos renderizados

En este apartado se muestra la comparativa entre las dos visualizaciones del gemelo digital, una desarrollada con el renderizado Estándar (véase 4.1.1) y la otra transformada con el renderizado de Alta Definición. Este análisis profundo analiza no solamente las diferencias visuales, sino que también el impacto significativo que la transición de un renderizado a otro puede tener en la calidad estética y la experiencia del usuario en un gemelo digital.

En esta comparativa, se define cómo la elección del renderizado influye en la iluminación, sombras, efectos visuales y otros aspectos clave de la representación gráfica del gemelo digital. A través de este proceso, se busca destacar las mejoras visuales evidentes y también comprender cómo estas transformaciones afectan la coherencia estética y la inmersión general en el entorno 3D.

- **Nave.** En primer lugar, se pueden ver en la Figura 4.14 que los reflejos y la iluminación en la nave es mucho más sofisticada y más semejante a los efectos de iluminación reales.



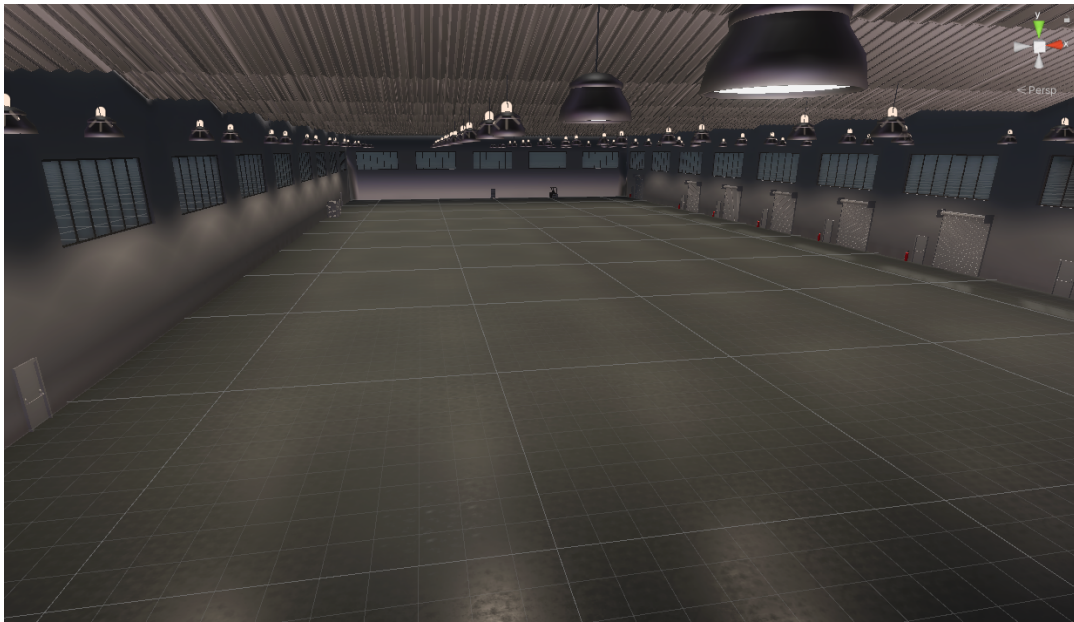
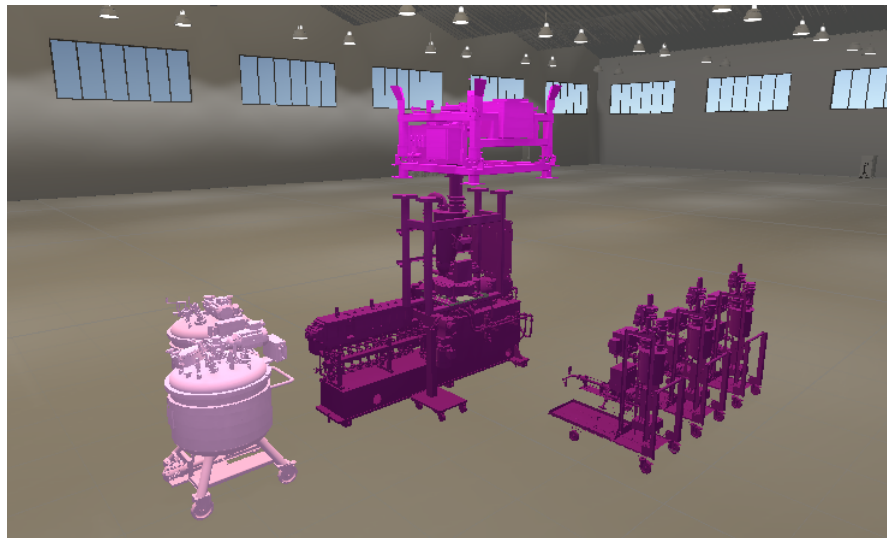


Figura 4.14: Comparativa de la nave con ambos renderizados. Renderizado Estándar arriba y renderizado en Alta Definición abajo. Fuente: Elaboración propia.

- **Extrusora.** Este es uno de los elementos donde más se puede apreciar los cambios en los elementos como se observa en la Figura 4.15. Las sombras están mejor definidos y los reflejos sobre los objetos están logrados. Asimismo, la definición de los materiales que lo conforman son notablemente superiores.



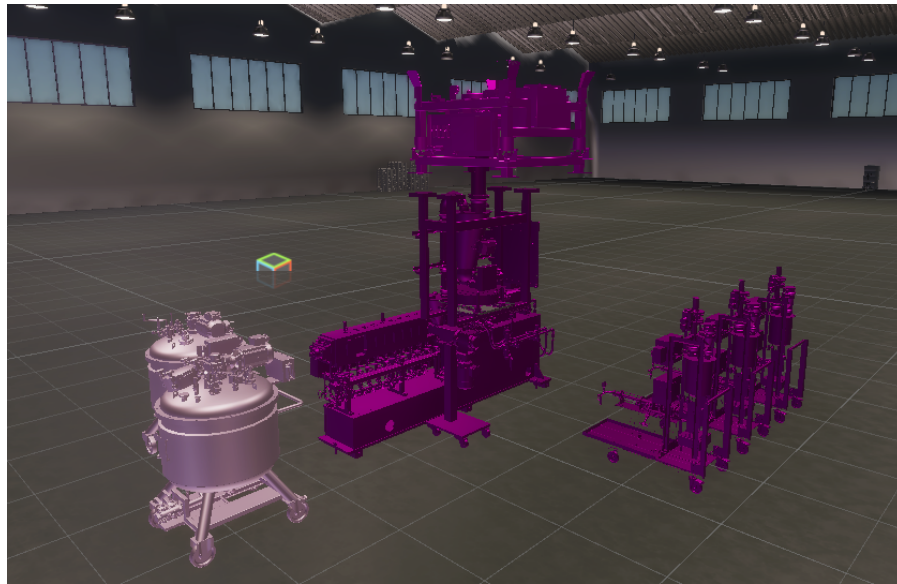
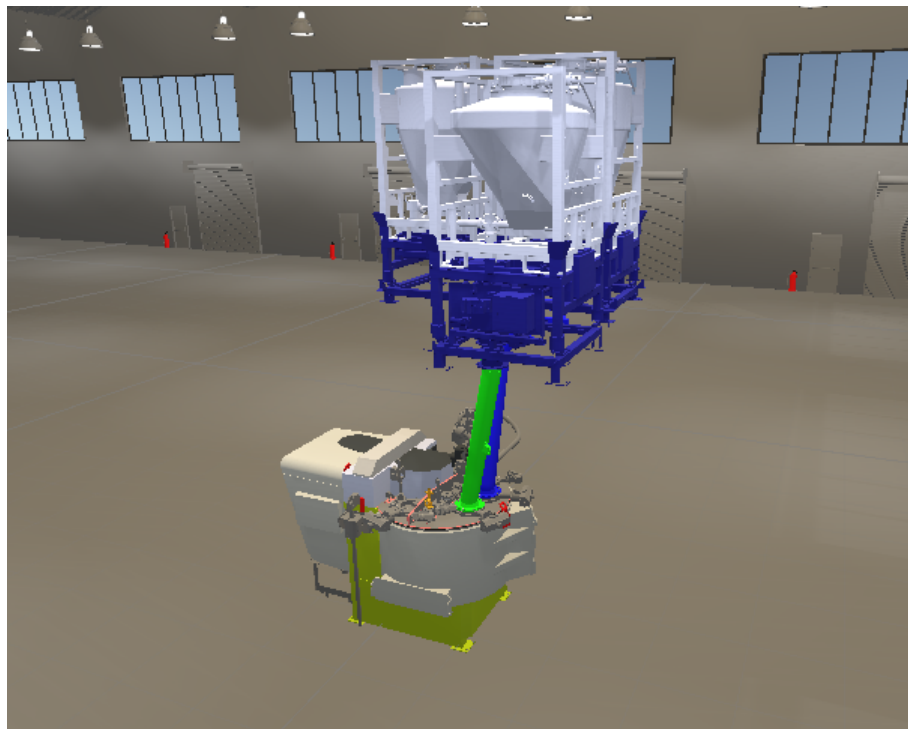


Figura 4.15: Comparativa de la extrusora con ambos renderizados. Renderizado Estándar arriba y renderizado en Alta Definición abajo. Fuente: Elaboración propia.

- **Mezcladora.** En este objeto, al tener varios materiales que lo conforman, son notables las diferencias entre ambos renderizados, en cuanto a sombras, iluminación y reflejos. Esto se ve representado en la Figura 4.16.



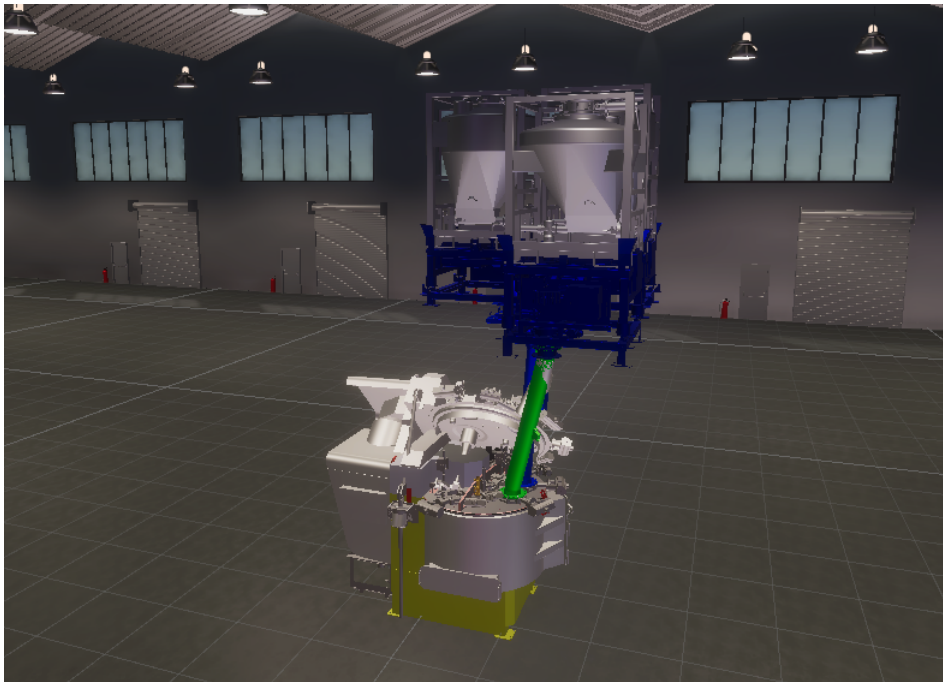
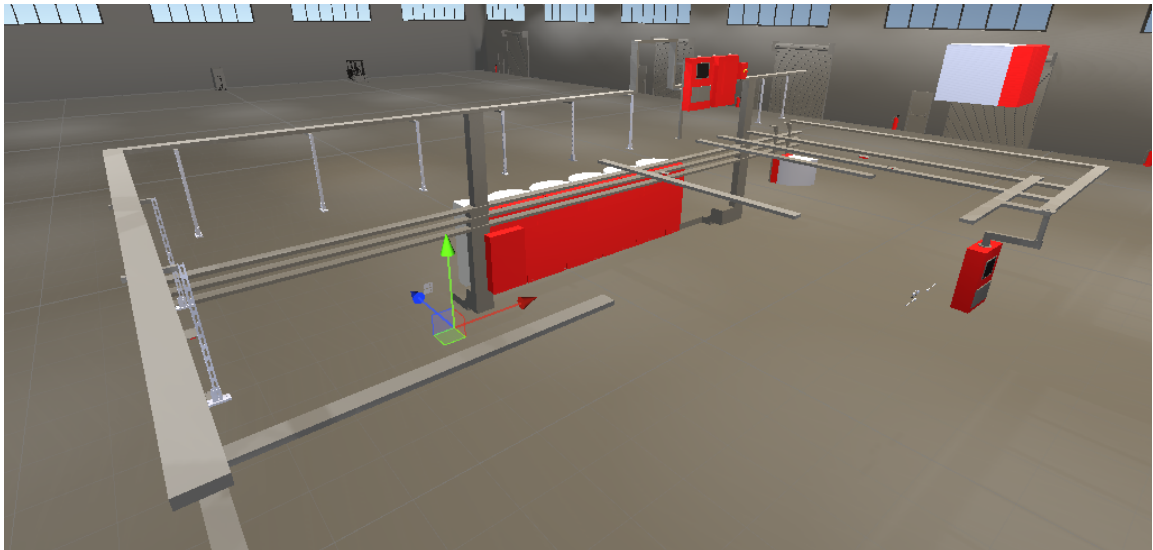


Figura 4.16: Comparativa de la mezcladora con ambos renderizados. Renderizado Estándar arriba y renderizado en Alta Definición abajo. Fuente: Elaboración propia.

- **Armario de control.** Esta estructura al tratarse de un objeto con colores más planos, la diferencia no es tan notable aunque es claro el salto de definición observable en la Figura 4.17.



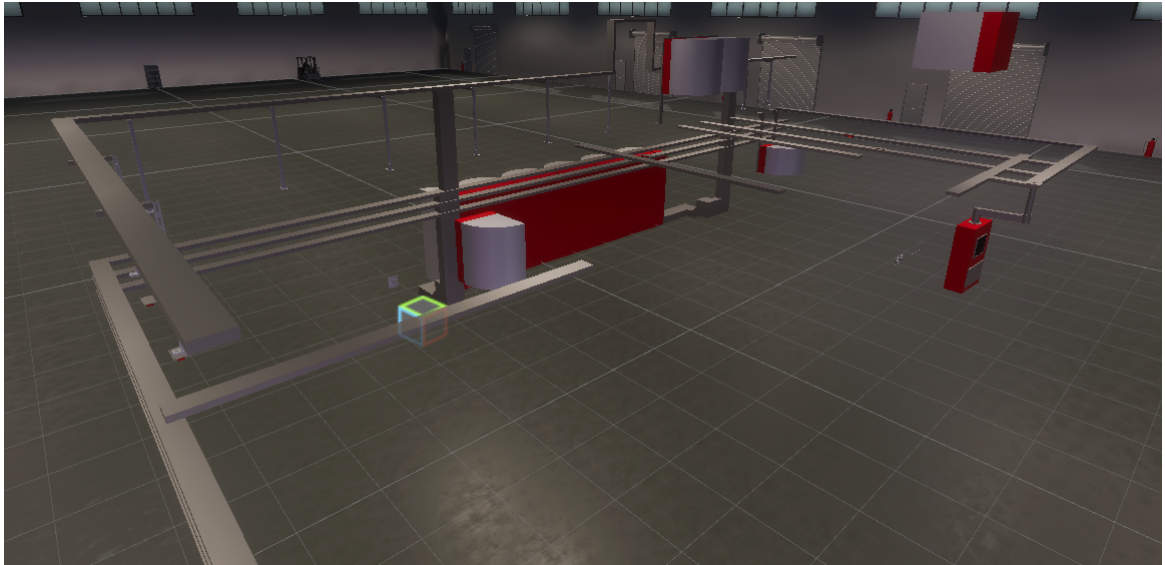


Figura 4.17: Comparativa del armario de control con ambos renderizados. Renderizado Estándar arriba y renderizado en Alta Definición abajo. Fuente: Elaboración propia.

- **Sacos big bag.** En este caso, se puede ver claramente en la Figura 4.18 como los efectos de iluminación afectan a este elemento. Al no recibir iluminación directa por parte de las luces de la nave, las sombras hacen un gran efecto sobre ella.

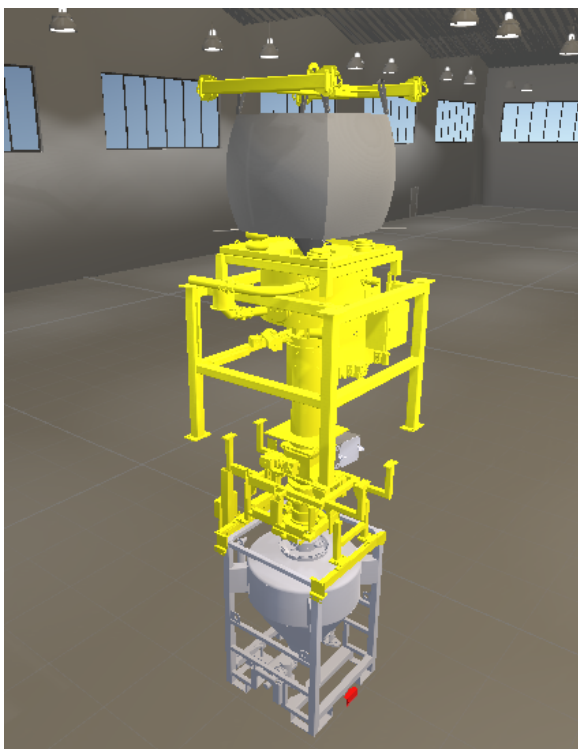


Figura 4.18: Comparativa del saco big bag con ambos renderizados. Renderizado Estándar izquierda y renderizado en Alta Definición derecha. Fuente: Elaboración propia.

- **Armario para bidones.** El armario para bidones es uno de los elementos donde su mejora no es tan evidente como en el resto, debido a que los relieves no son tan notables. Aún así la definición de los elementos es mejor como se ve en la Figura 4.19.

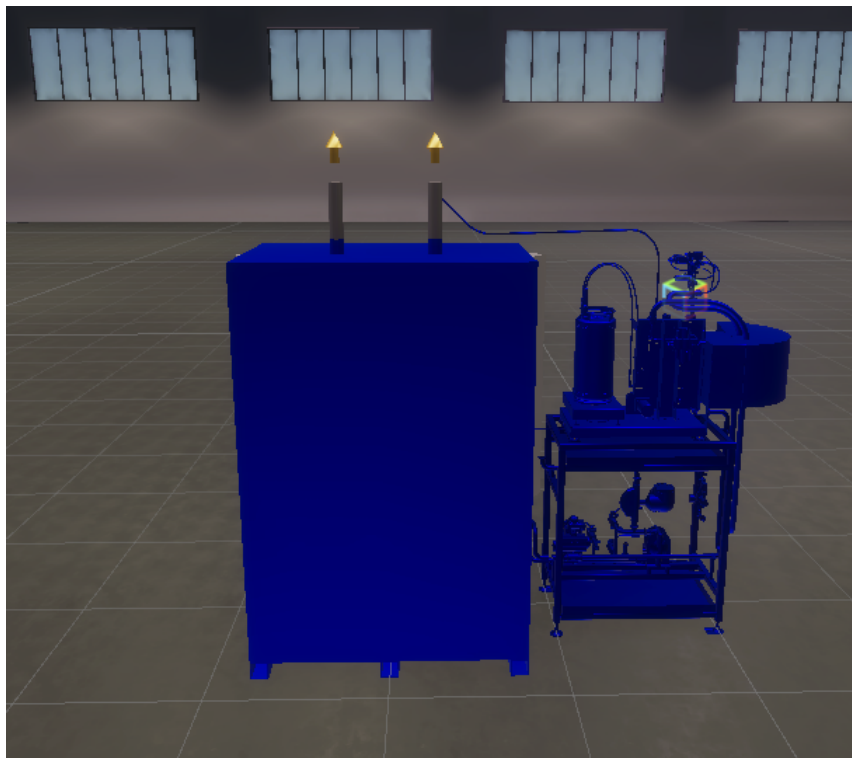
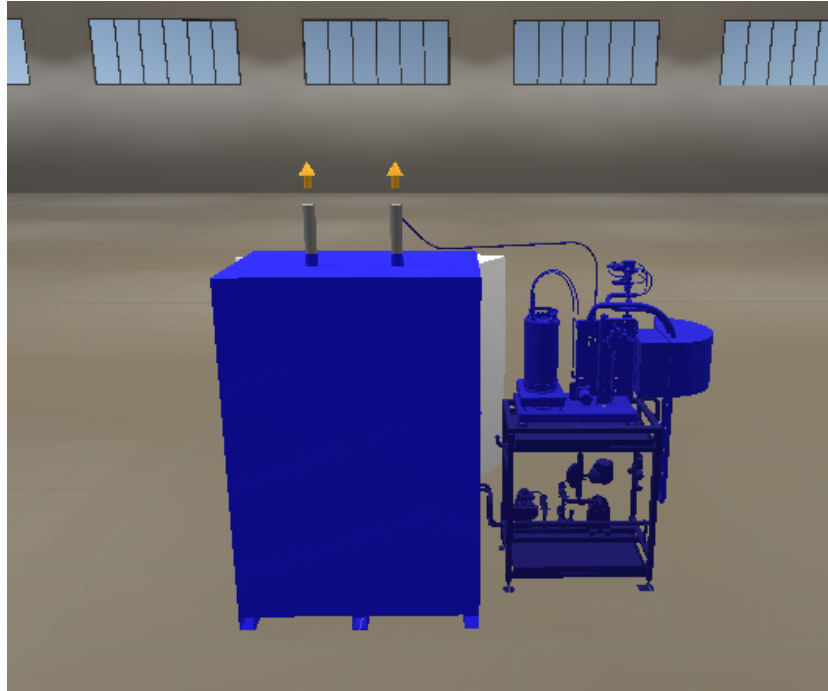


Figura 4.19: Comparativa del armario para bidones con ambos renderizados. Renderizado Estándar arriba y renderizado en Alta Definición abajo. Fuente: Elaboración propia.

- Caja de guantes.** La caja de guantes es el elemento que mayor salto de calidad recibe. Al estar conformado por una mayor cantidad de elementos, la mejora de la definición, precisión y sombras en la estructura es sobresaliente como se observa en la Figura 4.20.

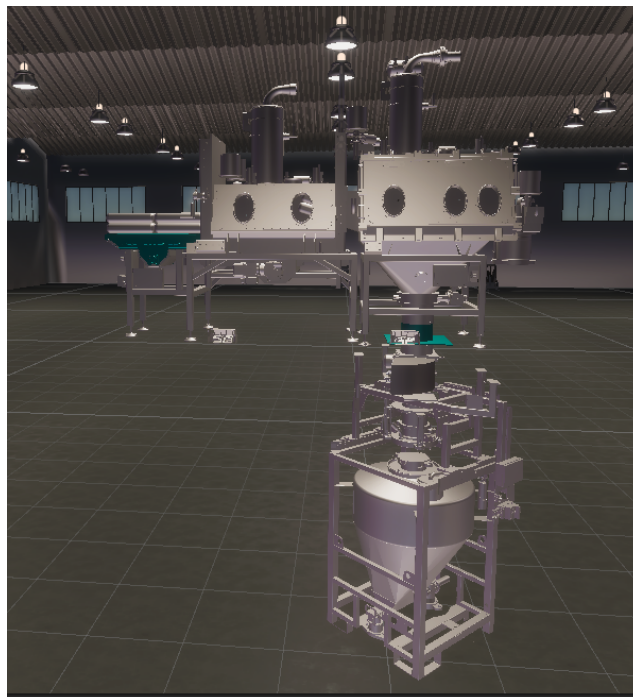
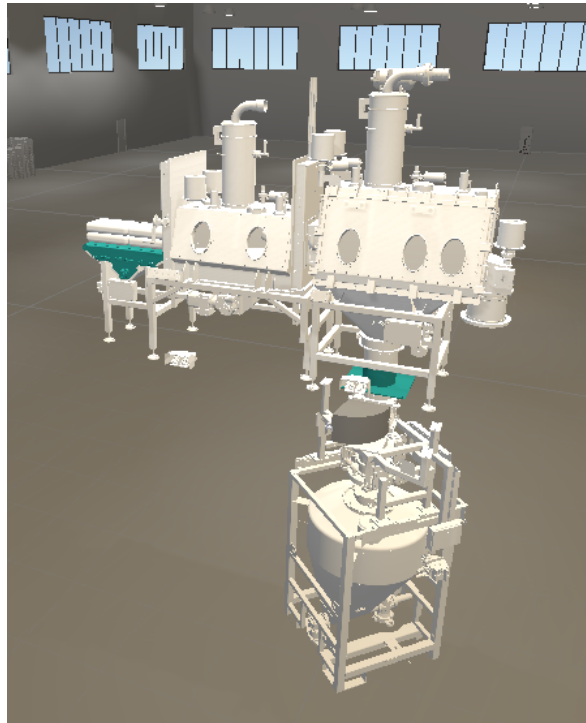


Figura 4.20: Comparativa de la caja de guantes con ambos renderizados. Renderizado Estándar arriba y renderizado en Alta Definición abajo. Fuente: Elaboración propia.

- **Grúa portátil.** Este el elemento sufre también la falta de iluminación directa por la posición en la que se encuentra respecto a las luces de la nave. Por ello, es la imagen es notablemente más oscura en la Figura 4.21, pero permite la visualización bien definida de como funcionan las sombras en este renderizado.

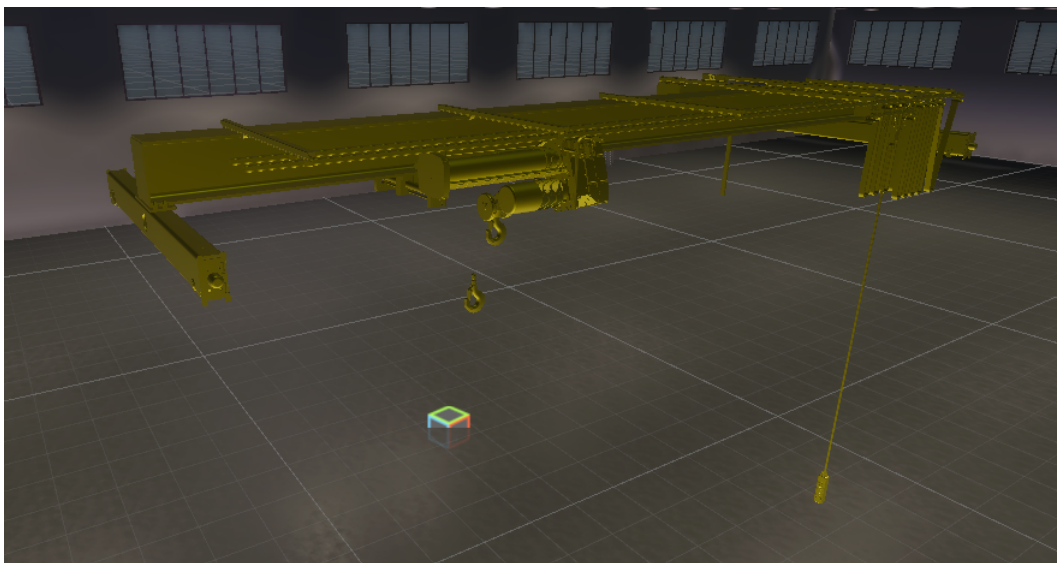
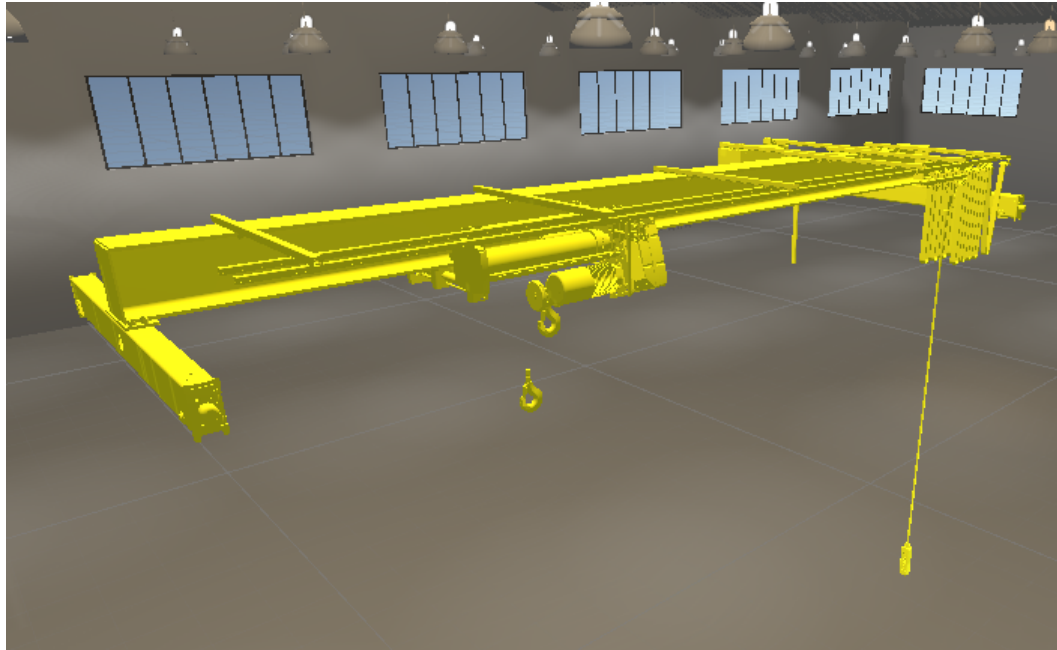


Figura 4.21: Comparativa de la grúa portátil con ambos renderizados. Renderizado Estándar arriba y renderizado en Alta Definición abajo. Fuente: Elaboración propia.

- **Bomba hidráulica.** En este caso, se observa en la Figura 4.22 una mejora considerable en cuanto a resolución y aspecto visual respecto a su formato en el renderizado Estándar.

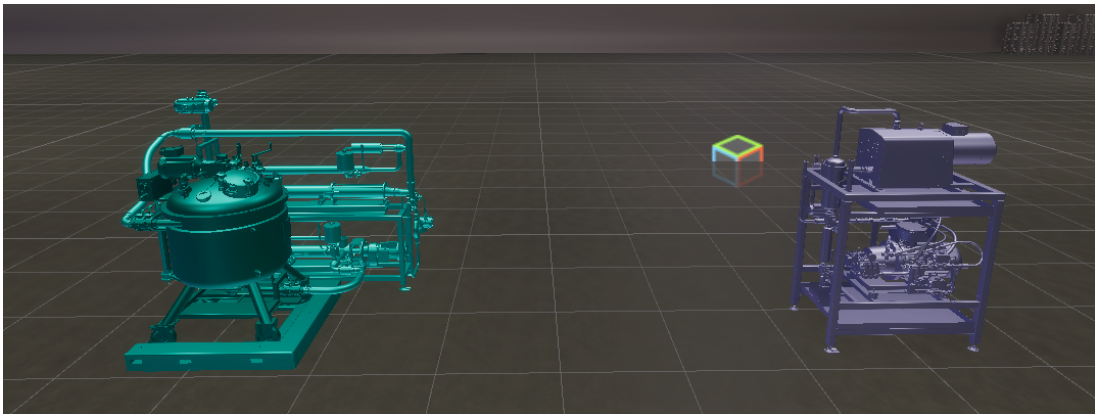
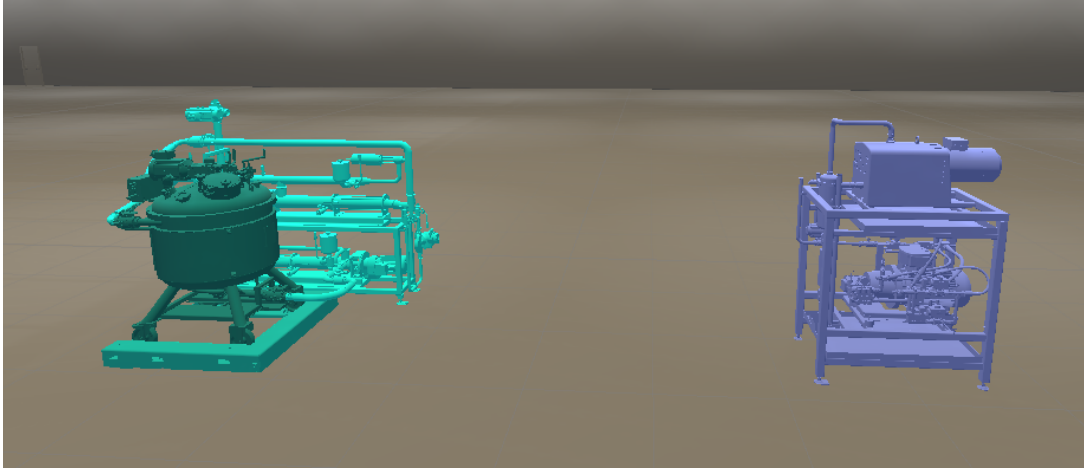


Figura 4.22: Comparativa de la bomba hidráulica con ambos renderizados. Renderizado Estándar arriba y renderizado en Alta Definición abajo. Fuente: Elaboración propia.

- **Descargador de sacos.** Estos objetos reflejan también el gran salto de calidad que da la representación del gemelo digital con el nuevo renderizado, observable en la Figura 4.23, debido a la gran variedad de materiales que lo conforman.

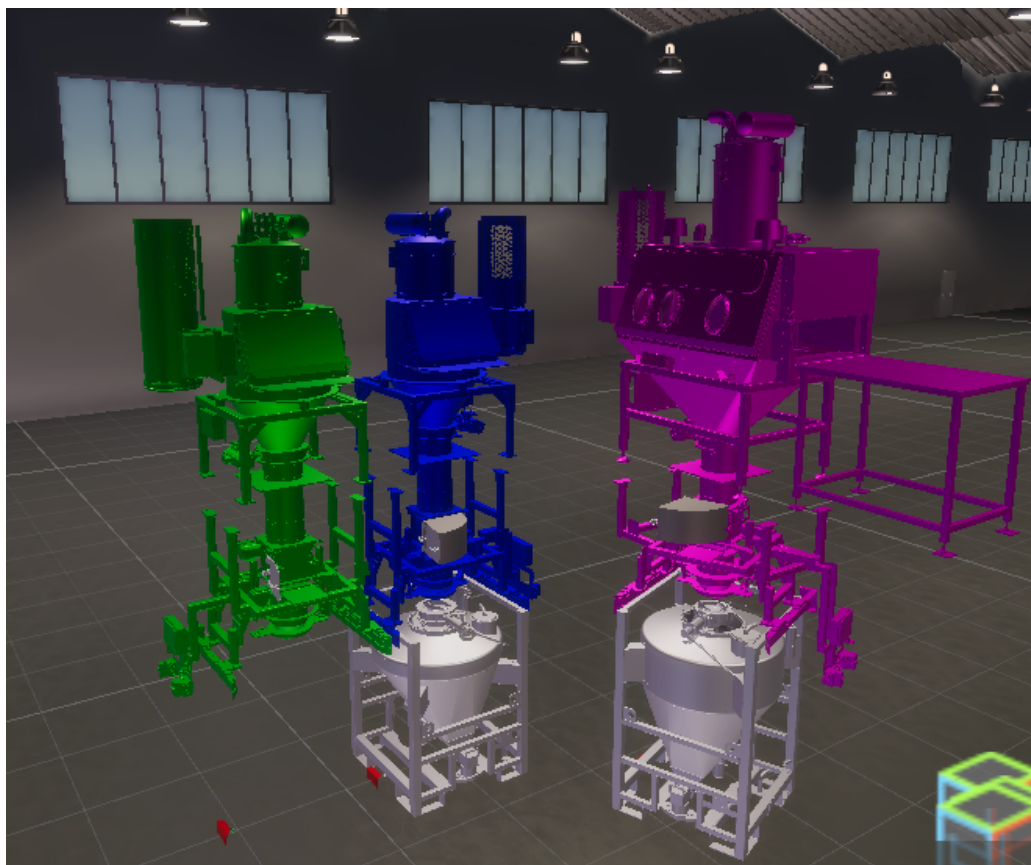
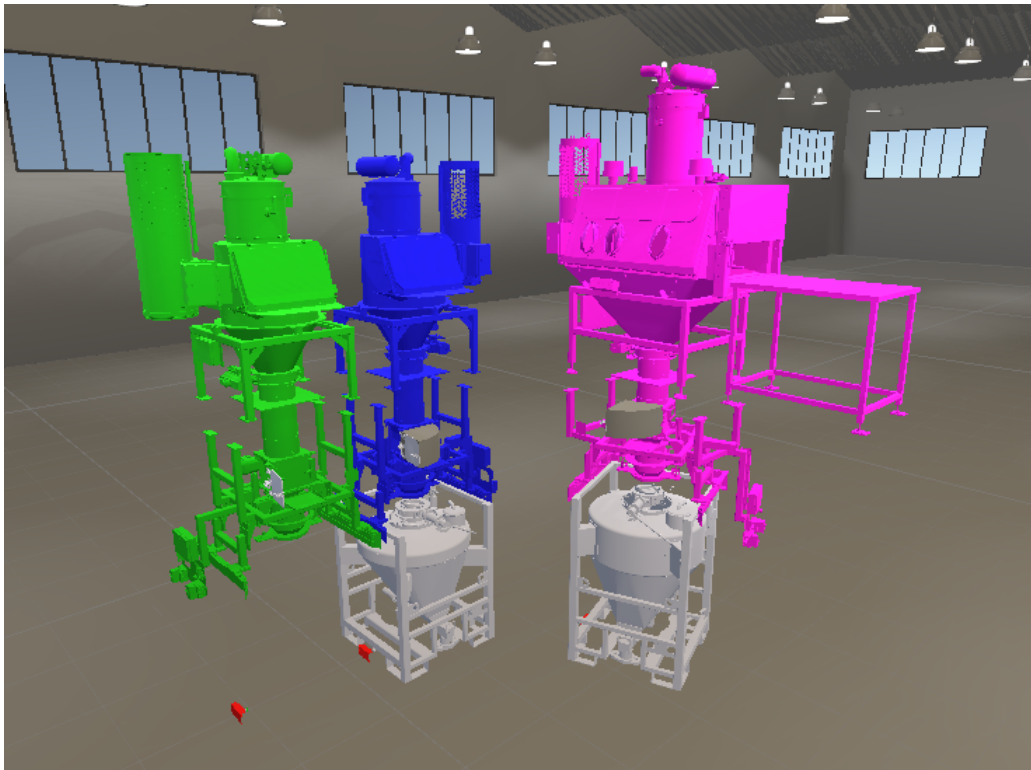


Figura 4.23: Comparativa del descargador de sacos con ambos renderizados. Renderizado Estándar arriba y renderizado en Alta Definición abajo. Fuente: Elaboración propia.

- **Transportador de vacío.** Al igual que el resto de los objetos, la definición de los materiales y los reflejos y sombras quedan mucho más definidas, como muestra la Figura 4.24.

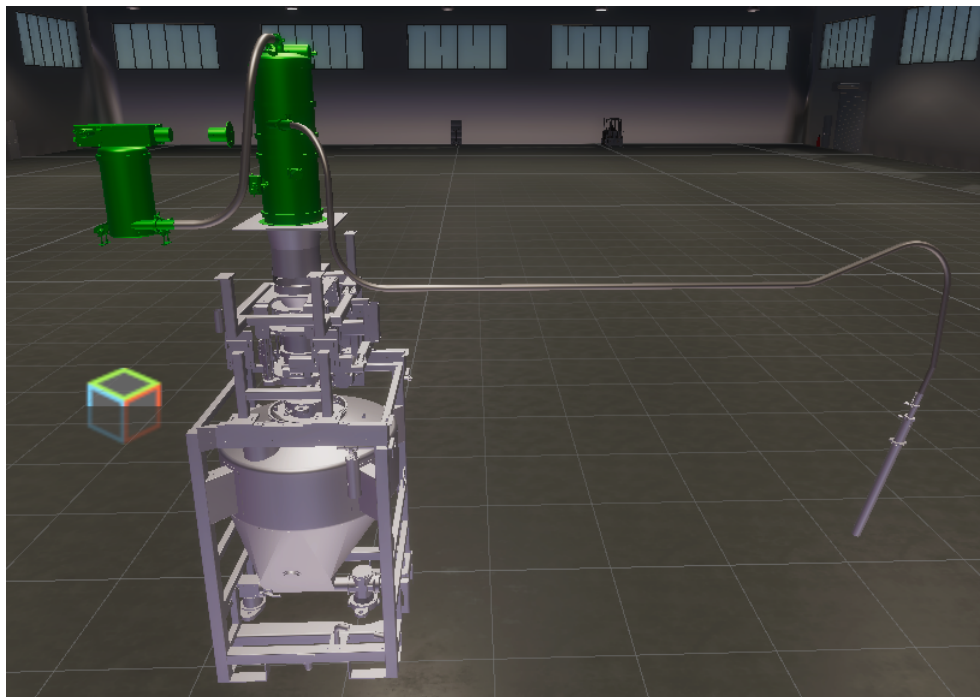
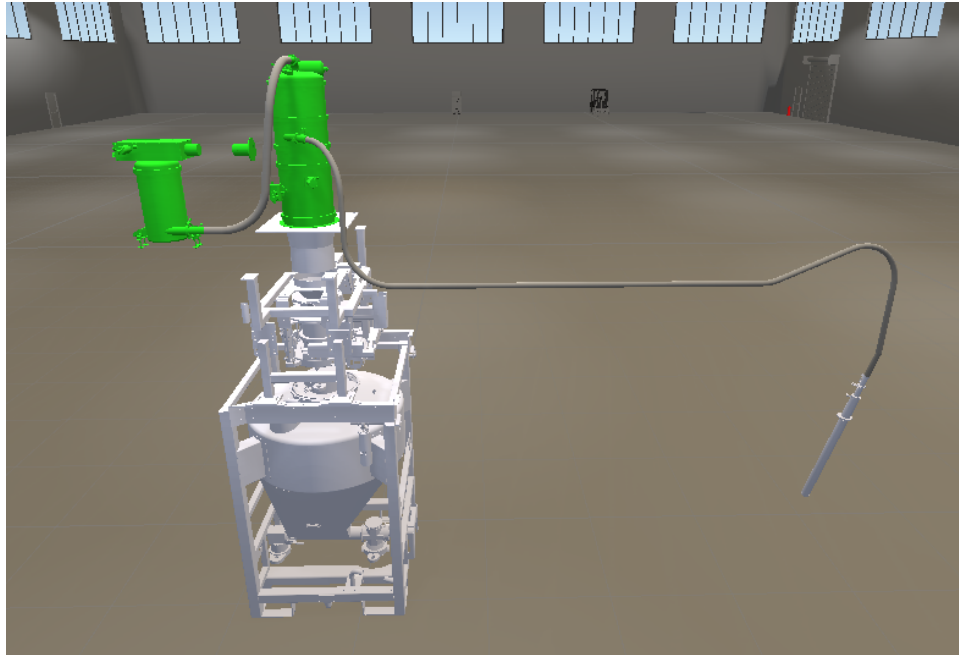


Figura 4.24: Comparativa del transportador de vacío con ambos renderizados. Renderizado Estándar arriba y renderizado en Alta Definición abajo. Fuente: Elaboración propia.

- Estructura de acero.** La Figura 4.25 contiene el elemento en el que mejor se aprecia mejor que en ningún otro cómo el renderizado de Alta Definición aplica con mejor precisión el impacto de la luz sobre los objetos y como estos aplican sombras y reflejos, semejantes a los de un elemento real.

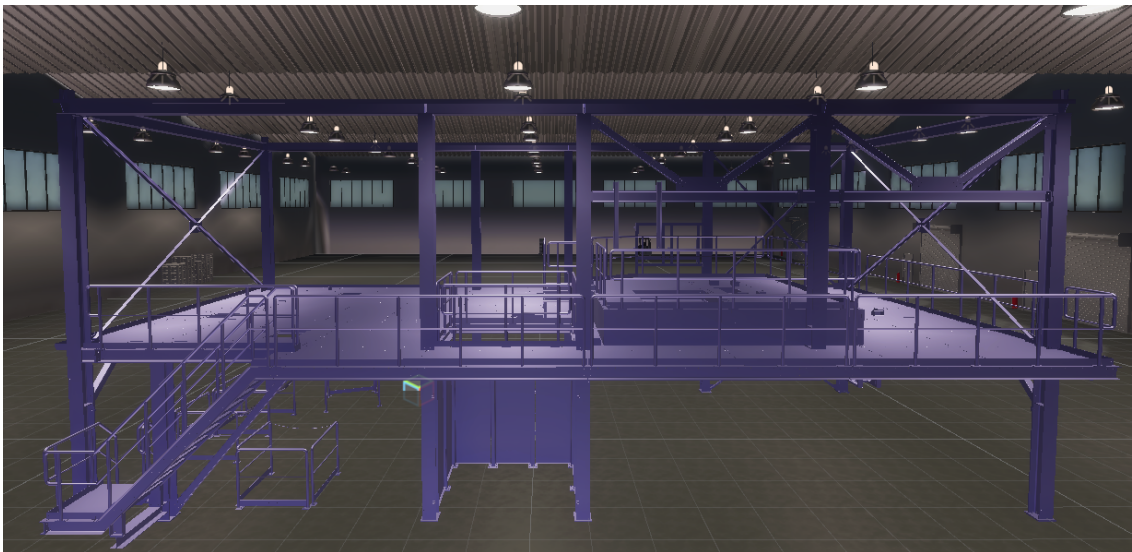
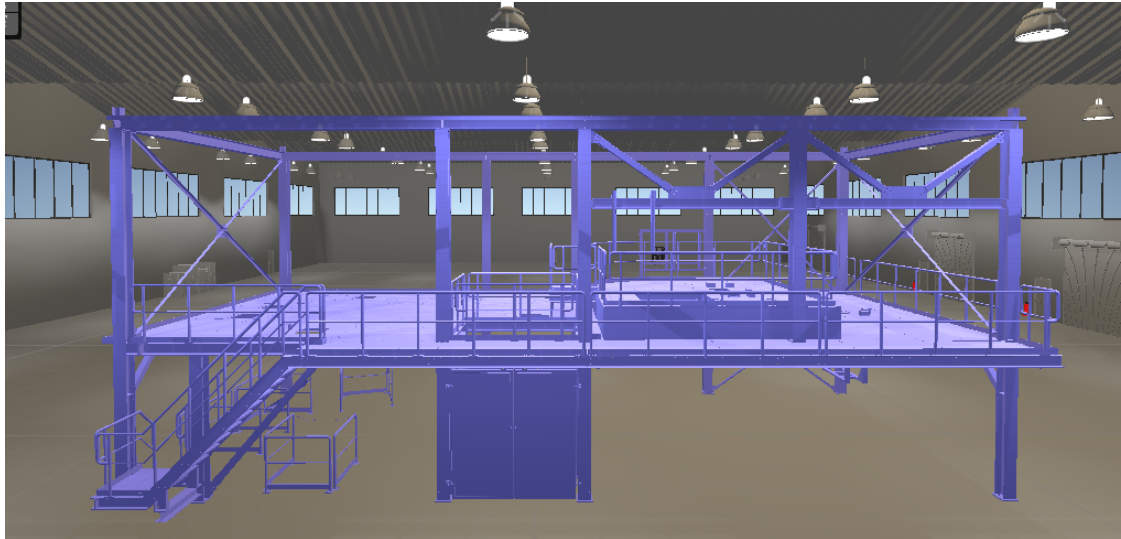


Figura 4.25: Comparativa de la estructura de acero con ambos renderizados. Renderizado Estándar arriba y renderizado en Alta Definición abajo. Fuente: Elaboración propia.

Una vez explicados y mostrados todos los componentes que conforman la representación visual del gemelo digital con el que se va a trabajar, la Figura 4.26 muestra la comparativa del conjunto completo.

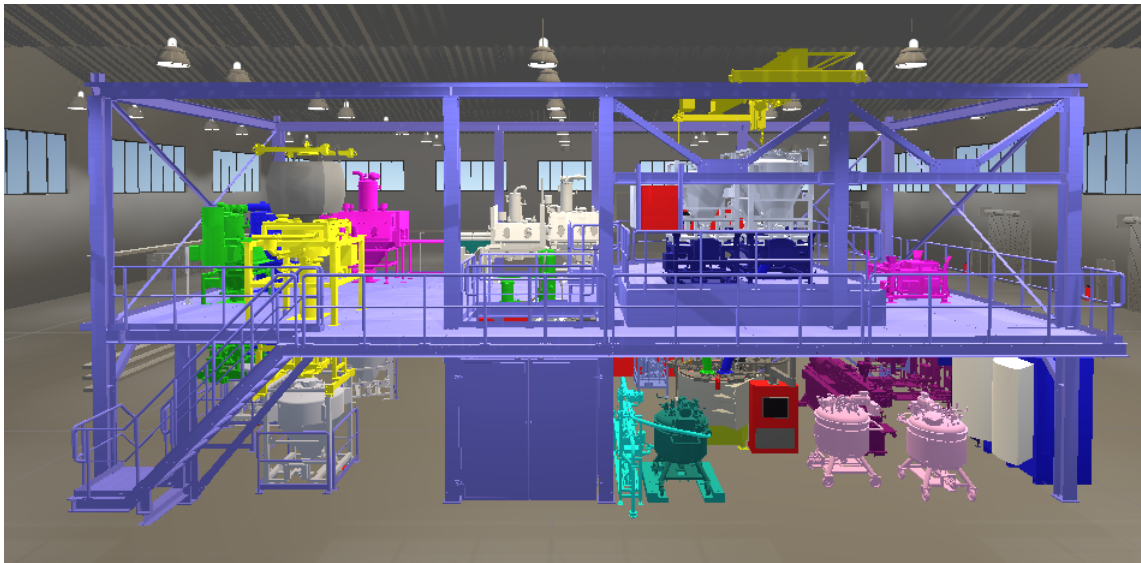


Figura 4.26: Comparativa del modelo completo con ambos renderizados. Renderizado Estándar arriba y renderizado en Alta Definición abajo. Fuente: Elaboración propia.

En resumen, esta comparativa pretende ofrecer una visión integral que sirva como evaluación técnica y como un ejercicio que influya en la toma de decisiones para futuros proyectos de desarrollo. Esta comparativa no solo resalta la evolución visual, claramente notoria, sino que también pone de manifiesto las consideraciones estratégicas y estéticas detrás de la elección de un renderizado de Alta Definición en la creación de gemelos digitales.

Capítulo 5

Funcionalidades de la visualización del gemelo digital en Alta Definición

5.1. Análisis de problemas con renderizado de Alta Definición

El salto evolutivo hacia el renderizado de Alta Definición en Unity ha introducido un panorama visual más sofisticado y estéticamente avanzado. Sin embargo, este avance no está exento de dificultades, y uno de los aspectos que ha destacado como área de complejidad dentro del entorno de este gemelo digital en la herramienta EDAS, es la función de resaltado, particularmente cuando se contrasta con su contraparte en el renderizado Estándar, donde sí funcionaba correctamente, como ya se mostró en el apartado 5.1.2.

La complejidad de la función de resaltado en el renderizado de Alta Definición radica en la sofisticación de los materiales y shaders asociados a los objetos. En comparación con el renderizado Estándar, los materiales y sombreadores en Alta Definición ofrecen una gama más amplia de propiedades y efectos visuales que ya se han expuesto, lo que enriquece la calidad estética pero también introduce una capa adicional de complejidad en la forma de tratar los objetos dentro de la escena y que a priori, como muestra la Figura 5.1, no se realiza el resaltado correctamente sino que simplemente se realiza un relleno del objeto seleccionado.



Figura 5.1: Visualización de la función resaltado en en Alta Definición. Fuente: Elaboración propia.

Además, surge un desafío adicional en la limitada disponibilidad de recursos de desarrollo para el renderizado de Alta Definición que puedan consultarse. Dada la relativa novedad de este sistema y la demanda de conocimientos especializados en el funcionamiento de sombreadores y programación avanzada, la búsqueda de soluciones específicas resulta más compleja, debido a que se pueden encontrar menos recursos y documentación comparados con entornos más consolidados. La combinación de enfrentarse a un nuevo programa y la falta de conocimiento de base de funciones y aplicaciones desarrolladas dentro del contexto del renderizado de Alta Definición se traduce en un desafío añadido para abordar este problema.

Este análisis se adentrará en la naturaleza de esta complejidad, examinando las diferencias principales de los materiales y shaders en ambos renderizados. Una vez comprendidas estas diferencias, se realizarán diversas propuestas para desarrollar funciones que permitan aprovechar estas características y poder resolver los inconvenientes en la función de resaltado.

5.1.1. Diferencias técnicas entre materiales y sombreadores en los renderizados Estándar y Alta Definición

Las diferencias técnicas entre los sistemas de renderizado Estándar y el renderizado de Alta Definición se manifiestan en la arquitectura subyacente de materiales y sombreadores, lo que incide directamente en la calidad y complejidad de la representación gráfica. Estas diferencias son fundamentales para comprender el salto cualitativo que supone la adopción de HDRP.

Renderizado Estándar

- **Simplicidad Estructural.** En el renderizado Built-in, los materiales y shaders son estructuralmente más simples y generalizados. Los shaders siguen modelos predefinidos, lo que simplifica el proceso de desarrollo pero limita la capacidad de lograr efectos visuales altamente detallados y realistas.
- **Limitaciones en Detalles Gráficos.** La capacidad para representar detalles gráficos, como reflejos avanzados, sombreado complejo o iluminación detallada, estaba restringida en comparación con sistemas más avanzados.

Renderizado de Alta Definición

- **Flexibilidad en materiales.** Introduce una arquitectura más flexible para materiales, permitiendo la creación de materiales altamente personalizados. Ofrece una amplia gama de propiedades y parámetros que pueden ajustarse con mayor precisión, permitiendo una representación visual más detallada.
- **Sombreadores avanzados.** Los sombreadores son más avanzados y permiten efectos visuales más realistas. Esto incluye capacidades mejoradas para la gestión de la luz, sombras, reflejos y refracciones, con un mayor grado de control sobre la interacción de la luz con los objetos.
- **Mayor complejidad y realismo.** Utiliza técnicas modernas de renderizado. Esto se traduce en una representación gráfica más fidedigna y estéticamente avanzada, con la capacidad de gestionar materiales más complejos y texturas detalladas.

En resumen, las diferencias técnicas radican en la capacidad del renderizado de Alta Definición para gestionar la complejidad a un nivel mucho más detallado que el renderizado Estándar. La adopción de implica una transformación técnica fundamental que permite una representación gráfica más avanzada y realista en los objetos.

5.1.2. Función de resaltado

Una función importante que contiene la herramienta EDAS (véase 1.2) cuando se hace uso de ella, es permitir la selección de objetos que se encuentran de la escena, en este caso el gemelo digital mostrado con anterioridad. Esta selección funciona de tal forma que al pinchar encima del objeto directamente en la visualización o en el desplegable donde se muestran los objetos que conforman la visualización del gemelo digital, se realiza un resaltado del borde del objeto deseado como muestra la Figura 5.2.



Figura 5.2: Visualización de la función resaltado en el entorno EDAS. Fuente: Elaboración propia.

Internamente, esta función cuando se selecciona un objeto, toma dicho objeto y los subobjetos asociados a él y recopila todos sus renderizadores de malla. Una vez hecho esto, la función asigna un material de contorno con un shader al objeto y sus subobjetos. Este sombreador es el encargado de realizar el contorno de los objetos a los que está asociado y darle el color asignado.

Esta función es importante, ya que permite mejorar la interfaz y mejora la experiencia del usuario mostrando información visual clave. Esto ofrece una realimentación inmediata cuando se selecciona el objeto deseado dentro de la aplicación. Asimismo, facilita la comprensión y la interacción para el usuario, ya que muestra qué objeto está activo y sobre el cual van a tener efecto las acciones que realice el usuario sobre él. Además permite enfocar la atención en un objeto o zona concreto sin verse afectado por el resto de objetos que conforman el gemelo digital.

5.2. Propuestas para solucionar los problemas

Una vez realizado este análisis, se plantean las propuestas para solucionar el problema del resaltado.

5.2.1. Solución a través del uso de activos de Unity

Como ya se comentó, la repositorio de activos de Unity (véase 2.3.1), ofrece la posibilidad de optar a recursos ofertados por parte de otros desarrolladores de Unity lo que permite reducir el tiempo de programación.

Para esta primera propuesta se utilizaron dos activos de la repositorio de activos:

- **Resaltado para Alta Definición.** Este activo, como su propio nombre adelanta, se trata de una herramienta que permite el resaltado de los objetos con características de Alta Definición.

La implementación de este activo requiere de una configuración previa en los ajustes del proyecto. Se modificaron los efectos de postprocesamiento, se introdujo un nuevo Volumen de Postprocesamiento y un nuevo material con un sombreador específico que proporciona el activo para realizar la función de resaltado. A continuación, se debe asociar a un objeto que tenga un renderizado de malla asociado. Una vez terminado, se crea un resaltado permanente.

- **Combinación de mallas.** Este activo permite combinar las mallas que conforman el objeto al que se asigna este activo y las mallas de sus subobjetos asociados. Este activo facilita un asistente que permite la implementación del mismo.

El uso del segundo activo es consecuencia de las características del primero. Al ser necesario que el objeto al que se asocie el activo de resaltado contenga una malla de renderizado, entra en conflicto con el funcionamiento de la función de resaltado en el renderizado Estándar (véase 5.1.2).

Esto es debido a que la función de resaltado inicial toma automáticamente los renderizados de malla de todos los subobjetos asociados al objeto principal, mientras que para este caso, el objeto principal tiene que tener un renderizado de malla propio. Para ello, se usa el segundo activo que permite agrupar todos los componente de malla de los subobjetos como un único renderizado de malla. Esta nueva malla se asocia al objeto principal y permite implementar el activo de resaltado.

Una vez hecho solucionado el problema del renderizado de malla, el siguiente paso es solucionar el momento en el que se active el resaltado, ya que se establece de manera permanente en la escena. Para ello, se edita la función ya creada que permite marcar los objeto y se añade el componente del activo dentro de dicha función, para que se active cuando se seleccione un objeto, y se desactive cuando se deseccione dicho objeto. Una vez se han realizado estos cambios el resultado obtenido se muestra en la Figura 5.3.

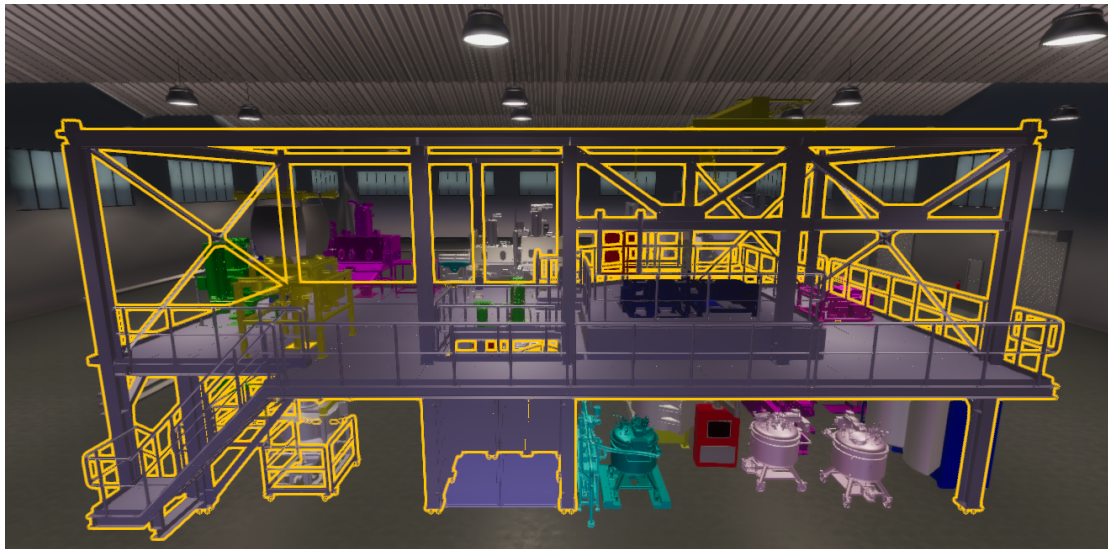


Figura 5.3: Visualización de la primera propuesta de función resaltado en Alta Definición. Fuente: Elaboración propia.

Sin embargo, a pesar de la gran apariencia que ofrece la solución propuesta, se ha descartado debido a la manipulación que hay que realizar sobre los objetos ya creados, dónde se han tenido que agrupar todos los componentes de un objeto en un único conjunto desde el modo Editor y por tanto, impide la selección de forma individual de los componentes en la herramienta EDAS.

Asimismo, el proceso de agrupación requiere de un consumo importante de tiempo y de recursos del programa, haciendo que se ralentice su funcionamiento. Por ello, esta resolución del problema no es conveniente.

La línea que se busca es similar a la obtenida con la función de resaltado inicial (véase 5.1.2), dónde la función obtenía todos los renderizados de malla automáticamente sin tener que agruparlas en una malla de gran tamaño. Es por ello, que la primera propuesta ha sido descartada.

5.2.2. Solución a través de los sombreadores de Unity

Debido a la resolución anterior, la segunda propuesta está enfocada en la idea de realizar la función de resaltado a través de la reestructuración de la programación de los sombreadores de Unity.

En este caso, la dificultad radica en que los materiales y los sombreadores en Alta Definición, como se ha matizado previamente, contienen componentes y estructuras internas de mayor complejidad que con el renderizado Estándar. El estudio de cómo modificar este tipo de programación es una tarea también complicada debido a que

la programación de shaders requiere unos conocimientos avanzados y de alto nivel de sombreadores.

Además, debido a esta dificultad, la información es más complicada de encontrar a través de internet, ya que no existen tantos expertos que dominen este área en el renderizado de Alta Definición.

A continuación se detalla brevemente la programación realizada para la solución mediante el uso de un shader. Dicha programación está presente en la Figura 5.4. En primer lugar se definen las propiedades del sombreador:

- **Color.** Define el color del resaltado de bordes.
- **Umbral.** Establece el umbral para determinar qué píxeles se consideran bordes. Valores más altos significan que se destacarán solo los bordes más pronunciados.
- **Anchura del borde.** Controla la anchura del resaltado del borde. Valores más altos dan bordes más anchos.

A continuación se inicia la sección de código en lenguaje de sombreado C para gráficos (Cg). En ella, se definen las estructuras para los datos de salida y entrada del vértice y se añade una función que transforma las coordenadas del vértice al espacio de proyección y establece la identificación de la instancia.

Por otro lado, se definen las características que van a tener el color de resaltado, el umbral para definir los bordes y su anchura. Se usan funciones para obtener la profundidad lineal del píxel actual (LinearEyeDepth) y para calcular la variación de profundidad (Fwidth).

La Figura 5.5 muestra la propuesta obtenida. Con esta solución no se obtiene el resultado esperado, ya que sigue mostrando el resaltado como al inicio de la conversión al renderizado de Alta Definición, donde en lugar de realizar un bordeado, se realiza un relleno de los componentes. Por ello, esta propuesta es descartada como solución, ya que requiere unos conocimientos más profundos que se escapan del contenido de este proyecto.

```

Shader "Custom/EdgeHighlightShader"
{
    Properties
    {
        _Color("Color", Color) = (1,1,0,1)
        _Threshold("Threshold", Range(0.1, 1)) = 0.5
        _EdgeWidth("Edge Width", Range(0, 0.3)) = 0.1
    }

    SubShader
    {
        Tags { "RenderType" = "Opaque" }
        Pass
        {
            CGPROGRAM
            #pragma vertex vert
            #pragma fragment frag

            #include "Packages/com.unity.render-pipelines.high-definition/Runtime/ShaderLibrary/HD.hlsl"

            struct appdata_t
            {
                float4 vertex : POSITION;
            };

            struct v2f
            {
                float4 vertex : SV_POSITION;
                UNITY_VERTEX_INPUT_INSTANCE_ID
            };

            v2f vert(appdata_t v)
            {
                v2f o;
                o.vertex = TransformObjectToHClip(v.vertex);
                UNITY_SETUP_INSTANCE_ID(v);
                return o;
            }

            fixed4 _Color;
            half _Threshold;
            half _EdgeWidth;

            fixed4 frag(v2f i) : SV_Target
            {
                float depth = LinearEyeDepth(SAMPLE_DEPTH_TEXTURE(_CameraDepthTexture, UNITY_PROJ_COORD(i.vertex)));
                float d = Fwidth(depth);
                half edgeFactor = saturate(d * _EdgeWidth * 1000);
                half rim = step(_Threshold, edgeFactor);
                return _Color * rim;
            }
            ENDCG
        }
    }
}
    
```

Figura 5.4: Código de la programación del sombreador. Fuente: Elaboración propia.

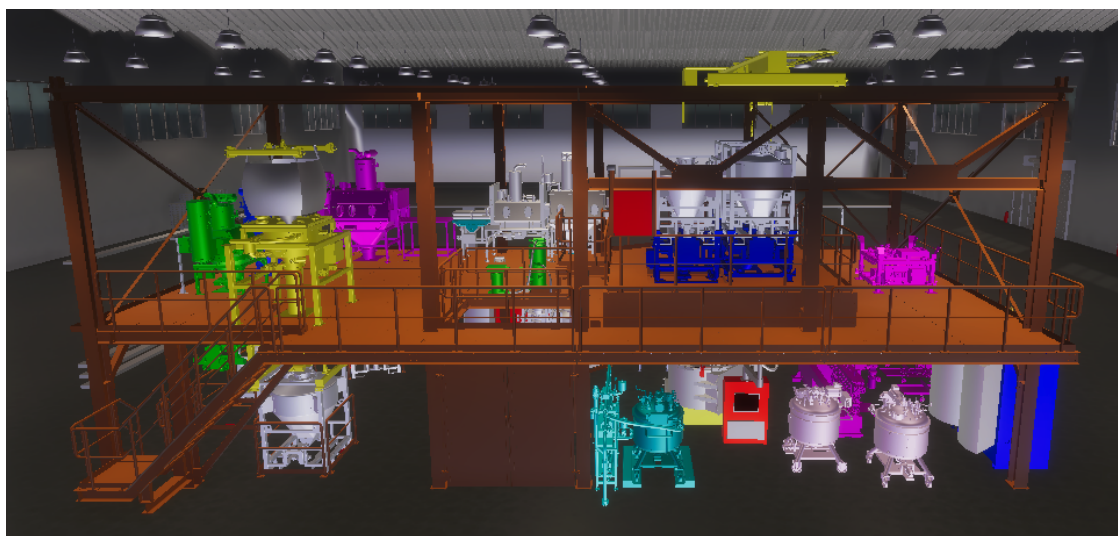


Figura 5.5: Visualización de la segunda propuesta de función resaltado en Alta Definición. Fuente: Elaboración propia.

5.2.3. Solución mediante la reprogramación de un activo

Para la última propuesta y por ende, la opción elegida, se ha seguido la ruta de trabajo de la primera propuesta e intentar realizar diversas modificaciones.

En esta propuesta, también se hace uso de los recursos de la repositorio de activos de Unity:

- **Resultado de fácil rendimiento 2D y 3D.** Este activo permite también realizar el resaltado de objetos para renderizado de Alta Definición.

Asimismo, ofrece un asistente que ayuda a realizar la configuración necesaria en cuanto a mapas de luz, sombras, máscaras y mallas y un nuevo volumen de renderizado, para poder ajustar de forma correcta el proyecto para el renderizado de Alta Definición.

Al terminar la configuración correctamente, es posible usar el activo. Este activo tiene dos partes principales, la primera es un script de programación que debe incluirse en la cámara de la escena. Este script ajusta las características de forma automática de la cámara para que se realice un correcto uso del activo y el resaltado funcione correctamente. Permite configurar diferentes parámetros como qué capas deben resaltarse, qué tipo de forma renderizado utilizar y la estrategia de renderizado que elegir. También se puede elegir qué tipos configuración elegir sobre el color de resaltado y la forma del mismo.

El segundo script, se añade a los objetos que se quieran resaltar dentro de la escena. Además, permite configurar el modo de dibujo, configura la capa de resaltado, también permite ajustar el estilo de renderizado y los ajustes de las mallas.



Figura 5.6: Visualización del primer procedimiento de la propuesta definitiva de resaltado en Alta Definición. Fuente: Elaboración propia.

Procedimientos dentro de la propuesta elegida

Este script al igual que la primera propuesta, realiza un resaltado permanente. Para realizar la modificación del resaltado y conseguir que se active durante la selección de objetos en lugar de forma continua, se ha dividido el trabajo en dos procedimientos

■ Comprobación del funcionamiento en el modo Editor

En este primer procedimiento se usa un script auxiliar ya existente y fácilmente manejable dentro de la ventana de edición con el fin de comprobar que el activo usado es viable para realizar la función de resaltado.

Este procedimiento consiste en realizar la técnica de resaltado a través de la activación de una variable externa dentro de la escena de Unity. De esta forma, se puede comprobar fácilmente que el activo realiza la función de resaltado de manera óptima, como muestra la Figura 5.6.

Una vez comprobado que el funcionamiento es correcto y que el activo se acopla a la idea propuesta, se comienza con el segundo procedimiento.

■ Implementación en la herramienta EDAS

El segundo procedimiento está relacionado con la implementación de la función de resaltado dentro de la aplicación EDAS cuando se está ejecutando. La complejidad de añadir el activo dentro de un script que se está ejecutando en tiempo de ejecución radica en cómo se debe tratar los objetos y las funciones asociadas a los mismo en ese script. Para ello se proponen dos soluciones:

```

public void MarkAsSelected(bool value)
{
    //Debug.Log(this.Name + " " + value);
    if (link)
    {
        // Cache renderers
        Renderer[] renderers = link.GetComponentsInChildren<Renderer>();
        EPOOutline.Outlinable outlineObject = link.GetComponent<EPOOutline.Outlinable>();

        switch (value)
        {
            case true:
                // Instantiate outline materials
                outlineObject.enabled = true;

                break;

            case false:
                // Remove outline shaders
                outlineObject.enabled = false;
                break;
        }

        if (link.TryGetComponent(out ISelectable selectable))
        {
            if (value) selectable.MarkAsSelected();
            else selectable.MarkAsUnselected();
        }
    }
}

```

Figura 5.7: Código de programación para referenciar al activo de resaltado en la herramienta EDAS. Fuente: Elaboración propia.

- **Activar las variables del anterior procedimiento.**

Esta solución resulta demasiado complicada debido a cómo se pueden tratar las señales entre scripts y cómo hacer referencias a ellas desde otro script. Además, estas variables están creadas a través de otro script, por lo que la referencia a dichas variables estaba encapsulada.

- **Añadir el activo al script que selecciona de elementos en EDAS.**

Con este método, se hace una referencia al activo que permite realizar el resaltado y activarlo cuando se seleccione el objeto deseado como se muestra en la Figura 5.7.

En primera instancia, este método funcionaba correctamente, pero generaba errores con ciertos componentes del gemelo digital. Este error era debido a que al seleccionar un objeto, dos de las funciones que usa el activo de resaltado toman el número de componentes de malla que tiene asociado ese objeto y los subobjetos asociados a él. Para algunos objetos, este número es superior a 1024 componentes de malla, que es el número máximo que permite Unity en el modo de ejecución para no ver comprometido el hardware de su API.

Este error era debido a que dos de las funciones que usa el activo de resaltado, manejaban un número mayor a 1024, que es el número que

permite Unity dentro del hardware de su API, para no comprometer el rendimiento del entorno.

Ese número está relacionado con el número de componentes de malla que tienen los objetos y sus respectivos subobjetos asociados. Al resultar que la mayoría de objetos tienen asociados una cantidad de componentes de malla superior al número que Unity permite manejar de una sola vez (1024), el resaltado para estos objetos no funcionaba correctamente, ya que Unity mostraba error.

La solución a este problema que surge, es añadir un bucle a ambas funciones que manejan la selección de los objetos, que permita dividir los componentes de los objetos en grupos de 1024 componentes de mallas para evitar ese problema, como muestran las Figuras 5.8 y 5.9, para posteriormente, aplicar el resaltado sin que se vea comprometido el rendimiento del entorno Unity ni entre en conflicto con los requisitos de hardware.

```
public static void Draw(OutlineParameters parameters, RenderTargetIdentifier target, RenderTargetIdentifier depth, Mat
{
    parameters.Buffer.SetRenderTarget(target, depth);
    if (viewport.HasValue)
        parameters.Buffer.SetViewport(viewport.Value);

    const int batchSize = 1023; // maximum number allowed per DrawMeshInstanced

    for (int i = 0; i < itemsToDraw; i += batchSize)
    {
        int batchCount = Mathf.Min(batchSize, itemsToDraw - i);
        Matrix4x4[] batchMatrices = new Matrix4x4[batchCount];
        Array.Copy(matrices, i, batchMatrices, 0, batchCount);

        if (SupportsInstancing)
            parameters.Buffer.DrawMeshInstanced(parameters.BlitMesh, 0, material, -1, batchMatrices, batchCount);
        else
            parameters.Buffer.DrawMesh(parameters.BlitMesh, Matrix4x4.identity, material, 0, -1);
    }
}
```

Figura 5.8: Código de programación para solucionar el error de limitación de 1024 componentes de malla en la función draw. Fuente: Elaboración propia.

```
public static void Blit(OutlineParameters parameters, RenderTargetIdentifier source, RenderTargetIdentifier destination, Ren
{
    var buffer = targetBuffer == null ? parameters.Buffer : targetBuffer;
    buffer.SetRenderTarget(destination, destinationDepth);
    if (viewport.HasValue)
        parameters.Buffer.SetViewport(viewport.Value);

    buffer.SetGlobalTexture(MainTexHash, source);

    const int batchSize = 1023; // maximum number allowed per DrawMeshInstanced

    for (int i = 0; i < itemsToDraw; i += batchSize)
    {
        int batchCount = Mathf.Min(batchSize, itemsToDraw - i);
        Matrix4x4[] batchMatrices = new Matrix4x4[batchCount];
        Array.Copy(matrices, i, batchMatrices, 0, batchCount);

        if (SupportsInstancing)
            buffer.DrawMeshInstanced(parameters.BlitMesh, 0, material, pass, batchMatrices, batchCount);
        else
            buffer.DrawMesh(parameters.BlitMesh, Matrix4x4.identity, material, 0, pass);
    }
}
```

Figura 5.9: Código de programación para solucionar el error de limitación de 1024 componentes de malla en la función blit. Fuente: Elaboración propia.

5.3. Validación

A continuación, la Figura 5.10 muestra el resultado de la función de resaltado para algunos objetos de la visualización del gemelo digital:

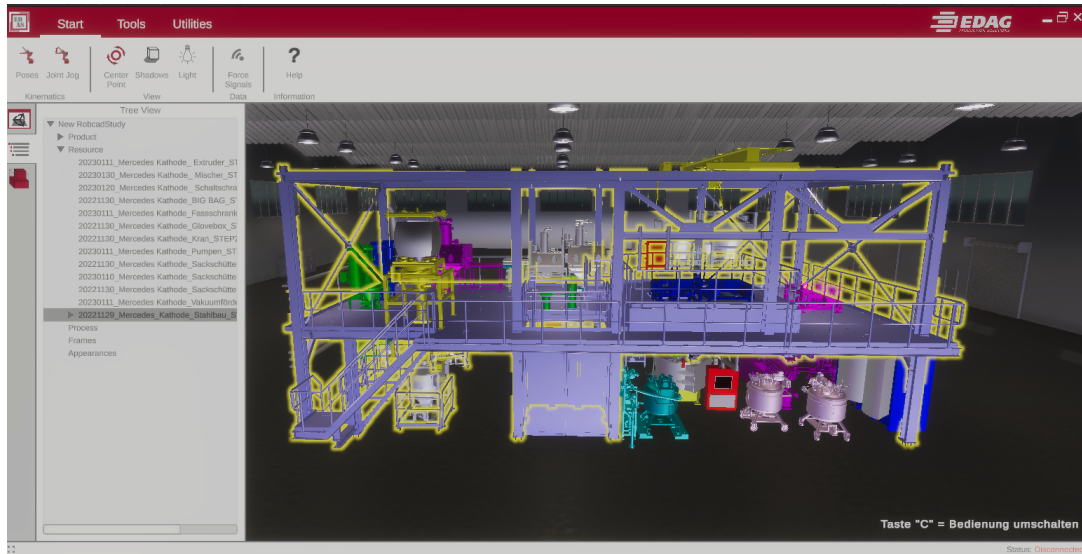




Figura 5.10: Visualización de la propuesta definitiva de resaltado en Alta Definición. Fuente: Elaboración propia.

Como se observa, el resaltado se activa y cumple su función cuando se selecciona un objeto dentro del árbol de objetos o cuando se pulsa sobre él dentro de la aplicación. Incluso se resalta las partes del objeto que se encuentran ocultas por otros objetos, por lo que la funcionalidad del mismo es correcta.

Esta es la opción elegida ya que cumple los requisitos previamente mencionados, tomar todos los subobjetos de un componente del gemelo digital y resaltar de forma óptima los componentes en el entorno de renderizado de Alta Definición. De esta forma, se consigue realizar que la función de resaltado se aplique de forma automática al realizar la conversión a Alta Definición, sin tener que realizar cambios cada vez que se utilice HDRP.

Capítulo 6

Conclusiones y Líneas de trabajo futuras

6.1. Conclusiones

Este trabajo de fin de grado ha explorado exhaustivamente la implementación y optimización del renderizado de Alta Definición, centrándose en su aplicación para el desarrollo de gemelos digitales. A través de un análisis meticuloso y comparativo entre los sistemas de renderizado Estándar y renderizado de Alta Definición, este estudio ha demostrado que este renderizado ofrece mejoras sustanciales en la calidad visual y estética, siendo un elemento crucial para la creación de gemelos digitales más realistas y detallados.

Una de las contribuciones más significativas de este trabajo ha sido la demostración de cómo el renderizado de Alta Definición mejora aspectos clave como la iluminación, texturización, y representación de sombras, lo que resulta en una experiencia visual más rica y envolvente. Este avance no solo enriquece la representación visual de los gemelos digitales, sino que también mejora la precisión y la utilidad de estos modelos en aplicaciones prácticas, como la simulación y el análisis de datos en tiempo real.

Además, se ha abordado y superado con éxito desafíos técnicos específicos asociados con el renderizado de Alta Definición, como la complejidad en la implementación de la función de resaltado. La solución propuesta no solo resuelve este problema, sino que también sienta las bases para futuras mejoras y ajustes en el sistema, demostrando la flexibilidad y la capacidad de adaptación del HDRP.

Sin embargo, este trabajo también reconoce las limitaciones y desafíos que presenta el renderizado de Alta Definición, especialmente en términos de exigencias de hardware y la curva de aprendizaje para su implementación efectiva. Estos factores

pueden representar barreras para su adopción en proyectos con recursos limitados o para desarrolladores menos experimentados.

En conclusión, este estudio ha demostrado que, a pesar de ciertos retos, el renderizado de Alta Definición es una herramienta poderosa y versátil para el desarrollo de la visualización de gemelos digitales. Se ha mostrado cómo puede mejorar significativamente la calidad y realismo de estos modelos, lo que tiene implicaciones importantes para su uso en una variedad de campos, desde la ingeniería y arquitectura hasta el entretenimiento y la educación. Este trabajo sienta una base sólida para futuras investigaciones y desarrollos en este campo, abriendo el camino para exploraciones más profundas y aplicaciones innovadoras de la tecnología de gemelos digitales.

6.2. Líneas de trabajo futuras

El enfoque de este trabajo ha sido uno de los primeros pasos para la adaptación de la visualización del entorno del gemelo digital en el entorno EDAS, pero esto no implica que la mejora del mismo haya llegado a su fin, sino que aún se pueden seguir explorando nuevas herramientas a través del renderizado de Alta Definición e incluso mejorar las incluidas dentro de este trabajo.

6.2.1. Mejora de la función de resaltado

En primer lugar, se propone realizar una exhaustiva revisión de la función de resaltado actual, identificando los puntos fuertes que aporta. Esto implica un análisis detallado de su desempeño en diversas condiciones y entornos, evaluando su capacidad para resaltar cualquier elemento de forma precisa. También se podría hacer un análisis el rendimiento de la computadora que lo soporta y el uso de recursos que pide para intentar reducirlo y convertirlo en un herramienta más eficiente.

La implementación de estas mejoras se puede centrar en aspectos técnicos y la manejabilidad de esta herramienta. Se pueden explorar interfaces de usuario más intuitivas, controles de fácil acceso y métodos de interacción que simplifiquen el proceso de diseño y manipulación. La integración de tecnologías emergentes, como la realidad virtual o aumentada, podría ser considerada para potenciar aún más la experiencia del usuario.

Este enfoque continuo de mejora se ajusta a la demanda constante de innovación y también respalda la evolución de las capacidades y expectativas de los usuarios en el ámbito del diseño 3D. Al comprometerse con la mejora de la función de resaltado y otras características del renderizado de Alta Definición, se sientan las bases para un entorno más accesible, intuitivo y eficiente en el ámbito del diseño tridimensional.

6.2.2. Resaltado mediante el uso de sombreadores

Si bien es cierto que el uso de sombreadores requiere un conocimiento extenso del manejo tanto de sombreadores como de programación del mismo, como ya se comentó en el presente proyecto (véase 5.1.1), se podría realizar con mayor detenimiento una investigación de cómo funcionan y cómo deben tratarse sus componentes para poder manejar la función de renderizado a través de la programación de shaders.

Esto va de la mano con la optimización de los recursos usados dentro de la aplicación, ya que la solución adoptada a lo largo de este trabajo requiere una cantidad de recursos bastante densa a pesar de que cumple con el objetivo del mismo. Esto significa que cualquier entorno no es apto para soportar la solución propuesta en este proyecto.

6.2.3. Uso de otros activos de Unity

Actualmente, la industria de los gemelos digitales dentro de Unity no está igualmente desarrollada como lo sí lo está la parte enfocada al desarrollo de videojuegos, que es el objetivo principal de Unity. Esto implica que la mayoría de mejoras y actualizaciones dentro de la repositorio de activos de Unity esté enfocada en solucionar problemas que se encuentran en la industria de los videojuegos y no tanto en la ingeniería y los gemelos digitales.

Debido a esto, aunque actualmente no exista un activo que realice la función de resaltado con precisión y sin tener que modificarlo, no implica que a lo largo del tiempo se incorpore un nuevo activo que permite dar una solución mejorada a la propuesta presentada en este proyecto o que permita darle un enfoque que no implique el uso de tantos recursos.

6.2.4. Manual de uso

La creación de un manual de uso específico para la función de resaltado puede aportar beneficios significativos para manejar la experiencia del usuario.

Claridad y Entendimiento

Un manual proporciona una guía estructurada y detallada sobre cómo utilizar la función de resaltado de manera efectiva. Explicar los conceptos importantes, los controles disponibles y los parámetros ajustables ayudaría a los usuarios a comprender completamente la funcionalidad evitando problemas de uso.

Optimización del Aprendizaje

Para usuarios nuevos o aquellos menos familiarizados con el entorno de renderizado de Alta Definición, un manual sirve como una herramienta de aprendizaje valiosa. Facilita la adquisición de habilidades necesarias para aprovechar al máximo la función de resaltado, acelerando la curva de aprendizaje y permitiendo una rápida integración en el flujo de trabajo. Esto ahorra tiempo y recursos.

Resolución de Problemas

Incluir secciones dedicadas a la resolución de problemas comunes y proporcionar soluciones a preguntas frecuentes contribuye a la eficiencia operativa y reduce la necesidad de asistencia externa. Esto garantiza que los usuarios obtengan resultados predecibles y coherentes, independientemente de la complejidad de la tarea o del contexto de aplicación.

Actualizaciones y nuevas características

Con el tiempo, las funciones del software pueden actualizarse o expandirse. Un manual de uso puede actualizarse para reflejar cambios en la funcionalidad y presentar nuevas características.

Referencias

- [1] Grupo EDAG. Edag group overview. Obtenido de: <https://www.edag.com/en/edag-group/the-company-edag/edag-group-an-overview>, 2023. [Último acceso: 05/11/2023].
- [2] Grupo EDAG. The company edag. Obtenido de: <https://www.edag.com/en/edag-group/the-company-edag>, 2023. [Último acceso: 05/11/2023].
- [3] EDAG Engineering Group AG. Interim consolidated financial statements h1 2023. Obtenido de: <https://www.edag.com/en/edag-group/investor-relations/financial-reports>, 2023. Informe financiero de la compañía EDAG Group AG de la primera mitad del año 2023. [Último acceso: 05/11/2023].
- [4] Grupo EDAG. Vehicle development. Obtenido de: <https://www.edag.com/en/services/vehicle-development>, 2023. [Último acceso: 05/11/2023].
- [5] Grupo EDAG. Services. Obtenido de: <https://www.trive.me/#services>, 2023. [Último acceso: 05/11/2023].
- [6] EDAG Aeromotive GmbH. Edag aeromotive. Obtenido de: <https://aeromotive.edag.com/en/>, 2020. [Último acceso: 05/11/2023].
- [7] Grupo EDAG. Production solutions. Obtenido de: <https://www.edag.com/en/services/production-solutions>, 2022. [Último acceso: 05/11/2023].
- [8] EDAG Production Solutions GmbH & Co. KG. Smart factory by edag production solutions. Obtenido de: <https://smartfactory.edag.com/en/>, 2020. [Último acceso: 05/11/2023].
- [9] SAP SE. ¿qué es la industria 4.0? Obtenido de: <https://www.sap.com/latinamerica/products/scm/industry-4-0/what-is-industry-4-0.html>, 2021. [Último acceso: 07/11/2023].
- [10] IEBS Digital School. Industria 4.0: Qué es, beneficios y ejemplos? Obtenido de: <https://www.iebschool.com/blog/industria-cuarta-revolucion-industrial-business-tech-logistica/>, 2023. [Último acceso: 07/11/2023].

- [11] REPSOL. Innovación digital para procesos reales. Obtenido de: <https://www.repsol.com/es/energia-futuro/tecnologia-innovacion/gemelos-digitales/index.cshtml>, 2023. [Último acceso: 08/11/2023].
- [12] IBM. ¿qué es un gemelo digital? Obtenido de: <https://www.ibm.com/es-es/topics/what-is-a-digital-twin>, 2023. [Último acceso: 08/11/2023].
- [13] Amazon Web Services. ¿qué es la tecnología de gemelos digitales? Obtenido de: <https://www.ibm.com/es-es/topics/what-is-a-digital-twin>, 2023. [Último acceso: 08/11/2023].
- [14] Wikipedia. Gemelo digital. Obtenido de: https://es.wikipedia.org/wiki/Gemelo_digital, 2023. [Último acceso: 08/11/2023].
- [15] SIEMENS Software. Gemelo digital. Obtenido de: <https://www.plm.automation.siemens.com/global/es/our-story/glossary/digital-twin/24465>, 2023. [Último acceso: 08/11/2023].
- [16] Geprom. Part of Telefónica Tech. El gemelo digital en la industria. Obtenido de: <https://www.geprom.com/ventajas-de-un-gemelo-digital-en-la-industria/>, 2022. [Último acceso: 09/11/2023].
- [17] Innovar o morir. Gemelos digitales: definición, tipos, aplicaciones y ejemplos. Obtenido de: <https://innovaromir.com/gemelos-digitales-definicion-tipos-aplicaciones-y-ejemplos/#Aplicaciones-de-gemelos-digitales>, 2023. [Último acceso: 09/11/2023].
- [18] Unity Support. Unity: ¿qué es y cómo funciona? Obtenido de: <https://support.unity.com/hc/es/articles/7642130833812-Unity-Qu%C3%A9-es-y-c%C3%B3mo-funciona->, 2023. [Último acceso: 26/11/2023].
- [19] Manual de Unity. Servicios de unity. Obtenido de: <https://docs.unity3d.com/es/530/Manual/UnityServices.html>, 2023. [Último acceso: 09/11/2023].
- [20] Unity Industry. Introducción a unity industrial, 2022. [Último acceso: 09/11/2023].
- [21] Unity Support. Unity: ¿qué es y cómo funciona? Obtenido de: <https://support.unity.com/hc/es/articles/7645361023892-Unity-Asset-Store-qu%C3%A9-es-y-c%C3%B3mo-funciona>, 2023. [Último acceso: 25/11/2023].
- [22] Manual de Unity. Renderizado de malla. Obtenido de: <https://docs.unity3d.com/es/2018.4/Manual/class-MeshRenderer.html>, 2023. [Último acceso: 25/11/2023].
- [23] Manual de Unity. Sombreadores. Obtenido de: <https://docs.unity3d.com/Manual/Shaders.html>, 2021. [Último acceso: 25/11/2023].

- [24] NVIDIA. Vertex shader. Obtenido de: <https://www.nvidia.com/en-us/drivers/feature-vertexshader/#:~:text=Vertex%20Shaders%20don't%20actually,a%20different%20position%20in%20space.>, 2020. [Último acceso: 25/11/2023].
- [25] Medium. Rasterización en unity. Obtenido de: <https://medium.com/shader-coding-in-unity-from-a-to-z/rendering-pipe-line-f0471aa0904b>, 2019. [Último acceso: 25/11/2023].
- [26] Manual de Unity. Fragment shader. Obtenido de: <https://docs.unity3d.com/510/Documentation/Manual/SL-VertexFragmentShaderExamples.html#:~:text=Fragment%20Shader%20is%20a%20shader,function%20of%20the%20fragment%20shader.>, 2021. [Último acceso: 25/11/2023].
- [27] Brakeza. Entendiendo el z-buffer. Obtenido de: <https://brakeza.com/entendiendo-el-z-buffer/>, 2020. [Último acceso: 01/12/2023].
- [28] Geekno. Culling. Obtenido de: <https://www.geekno.com/glosario/culling>, 2019. [Último acceso: 01/12/2023].
- [29] Manual de Unity. Visión general del postprocesamiento. Obtenido de: <https://docs.unity3d.com/es/2021.1/Manual/PostProcessingOverview.html>, 2021. [Último acceso: 01/12/2023].
- [30] Manual de Unity. Using the built-in render pipeline. Obtenido de: <https://docs.unity3d.com/es/2021.1/Manual/built-in-render-pipeline.html>, 2021. [Último acceso: 16/11/2023].
- [31] Manual de Unity. Extending the built-in render pipeline with commandbuffers. Obtenido de: <https://docs.unity3d.com/Manual/GraphicsCommandBuffers.html#:~:text=A%20CommandBuffer%20holds%20a%20list,and%20extend%20Unity's%20rendering%20functionality.>, 2021. [Último acceso: 16/11/2023].
- [32] Manual de Unity. Rendering paths in the built-in render pipeline. Obtenido de: <https://docs.unity3d.com/es/2021.1/Manual/RenderingPaths.html>, 2021. [Último acceso: 18/11/2023].
- [33] Manual de Unity. Forward rendering path. Obtenido de: <https://docs.unity3d.com/es/2021.1/Manual/RenderTech-ForwardRendering.html>, 2021. [Último acceso: 18/11/2023].
- [34] Manual de Unity. Deferred shading rendering path. Obtenido de: <https://docs.unity3d.com/es/2021.1/Manual/RenderTech-DeferredShading.html>, 2021. [Último acceso: 18/11/2023].
- [35] Manual de Unity. Vertex lit rendering path. Obtenido de: <https://docs.unity3d.com/es/2021.1/Manual/RenderTech-VertexLit.html>, 2021. [Último acceso: 18/11/2023].

- [36] Manual de Unity. Rendering order in the built-in render pipeline. Obtenido de: <https://docs.unity3d.com/Manual/built-in-rendering-order.html>, 2021. [Último acceso: 19/11/2023].
- [37] Unity. Universal render pipeline (urp). Obtenido de: <https://unity.com/srp/universal-render-pipeline>, 2021. [Último acceso: 29/11/2023].
- [38] Medium. Unveiling the power of urp in unity: A comprehensive guide. Obtenido de: <https://medium.com/@be.content23/unveiling-the-power-of-urp-in-unity-a-comprehensive-guide-7dbe8cbfcdcd>, 2023. [Último acceso: 29/11/2023].
- [39] Manual de Unity. Universal render pipeline overview. Obtenido de: <https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal@17.0/manual/index.html>, 2021. [Último acceso: 29/11/2023].
- [40] NoBreakPoint. Einführung zur high-definition render pipeline. Obtenido de: <https://blog.nobreakpoints.com/unity-hdrp-einfuehrung/#elementor-toc...heading-anchor-10>, 2023. [Último acceso: 16/11/2023].
- [41] Unity. About shader graph. Obtenido de: <https://docs.unity3d.com/Packages/com.unity.shadergraph@16.0/manual/index.html>, 2023. [Último acceso: 16/11/2023].
- [42] Educative. What is a shader graph in unity? Obtenido de: <https://www.educative.io/answers/what-is-a-shader-graph-in-unity>, 2022. [Último acceso: 16/11/2023].
- [43] Unity. Antialiasing in the high definition render pipeline. Obtenido de: <https://docs.unity3d.com/Packages/com.unity.render-pipelines.high-definition@14.0/manual/Anti-Aliasing.html>, 2022. [Último acceso: 16/11/2023].
- [44] Unity. Anti-aliasing, volume, and exposure for pc and console games. Obtenido de: <https://unity.com/how-to/anti-aliasing-volume-exposure-with-hdrp>, 2022. [Último acceso: 16/11/2023].
- [45] Unity Manual. Differences between forward and deferred rendering in hdrp. Obtenido de: <https://docs.unity3d.com/Packages/com.unity.render-pipelines.high-definition@14.0/manual/Forward-And-Deferred-Rendering.html#differences-between-forward-and-deferred-rendering-in-hdrp>, 2022. [Último acceso: 16/11/2023].
- [46] VionixStudio. Urp vs hdrp in unity. Obtenido de: <https://vionixstudio.com/2022/02/12/urp-vs-hdrp-in-unity/>, 2023. [Último acceso: 25/11/2023].