



UNIVERSIDAD
CATÓLICA
DE CUENCA

UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

**FACULTAD DE INFORMÁTICA, CIENCIAS DE LA
COMPUTACIÓN E INNOVACIÓN TECNOLÓGICA**

CARRERA DE REALIDAD VIRTUAL Y VIDEOJUEGOS

**Modelado, texturizado y animación de personajes y escenarios para el
videojuego “InkFall”**

**PROYECTO DE INTEGRACIÓN CURRICULAR PREVIO A LA
OBTENCIÓN DEL TÍTULO DE INGENIERO EN REALIDAD
VIRTUAL Y VIDEOJUEGOS**

AUTOR: JUAN JOSÉ BACUILIMA PULLA.

DIRECTOR: ING. JOHN LUIS ESTRADA GUAYTA, M.S.

CUENCA- ECUADOR

2026

DIOS, PATRIA, CULTURA Y DESARROLLO



UNIVERSIDAD CATÓLICA DE CUENCA
Comunidad Educativa al Servicio del Pueblo

**FACULTAD DE INFORMÁTICA, CIENCIAS DE LA
COMPUTACIÓN E INNOVACIÓN TECNOLÓGICA**
CARRERA DE REALIDAD VIRTUAL Y VIDEOJUEGOS

**Modelado, texturizado y animación de personajes y escenarios para el
videojuego “InkFall”**

**PROYECTO DE INTEGRACIÓN CURRICULAR PREVIO A LA
OBTENCIÓN DEL TÍTULO DE INGENIERO EN REALIDAD
VIRTUAL Y VIDEOJUEGOS**

AUTOR: JUAN JOSÉ BACUILIMA PULLA.
DIRECTOR: ING. JOHN LUIS ESTRADA GUAYTA, M.S.

CUENCA- ECUADOR
2026

DIOS, PATRIA, CULTURA Y DESARROLLO



Declaratoria de Autoría y Responsabilidad

Juan José Bacuilima Pulla portador de la cédula de ciudadanía N° **0150558070**. Declaro ser el autor de la obra: **Modelado, texturizado y animación de personajes y escenarios para el videojuego "InkFall"**, sobre la cual me hago responsable sobre las opiniones, versiones e ideas expresadas. Declaro que la misma ha sido elaborada respetando los derechos de propiedad intelectual de terceros y eximo a la Universidad Católica de Cuenca sobre cualquier reclamación que pudiera existir al respecto. Declaro finalmente que mi obra ha sido realizada cumpliendo con todos los requisitos legales, éticos y bioéticos de investigación, que la misma no incumple con la normativa nacional e internacional en el área específica de investigación, sobre la que también me responsabilizo y eximo a la Universidad Católica de Cuenca de toda reclamación al respecto.

Cuenca, **07 de septiembre del 2026**

F: 

Juan José Bacuilima Pulla

C.I. 0150558070



Certificado

Certifico que el presente trabajo de investigación fue desarrollado por **Juan José Bacuilima Pulla**, portador(a) de la cedula de ciudadanía N.º **0150558070** con el tema **Modelado, texturizado y animación de personajes y escenarios para el videojuego "InkFall"**, bajo mi supervisión.

Ing. John Luis Estrada Guayta, Ms.

DIRECTOR DEL TRABAJO INVESTIGATIVO

UNIVERSIDAD CATÓLICA DE CUENCA

Dedicatoria

Dedico este trabajo, en primer lugar, a Dios y a mi familia, por ser una fuente constante de apoyo, motivación y fortaleza a lo largo de mi formación académica. Su confianza y acompañamiento han sido fundamentales para continuar avanzando y superar los diferentes desafíos presentados durante este proceso.

A mis padres, por su esfuerzo, paciencia y por impulsarme siempre a seguir adelante y alcanzar mis metas. Cada logro obtenido durante esta etapa también representa el resultado de su apoyo y de los valores que me han transmitido.

Finalmente, dedico este trabajo a todas aquellas personas que estuvieron presentes durante este camino y que, de una u otra manera, contribuyeron a que pudiera culminar una etapa tan importante de mi vida.

Agradecimiento

Agradezco principalmente a Dios y a mi familia por acompañarme durante todo mi proceso de formación universitaria, por brindarme su apoyo en los momentos difíciles y por motivarme constantemente a continuar trabajando para alcanzar mis objetivos.

Expreso también mi agradecimiento a la Universidad Católica de Cuenca y a los docentes de la carrera de Ingeniería en Realidad Virtual y Videojuegos, quienes, a través de sus conocimientos y enseñanzas, contribuyeron al desarrollo de las habilidades profesionales y técnicas que fueron necesarias para la realización de este proyecto.

De manera especial, agradezco a mi tutor de tesis, John Luis Estrada Guayta, M.S., por su orientación, recomendaciones y acompañamiento durante el desarrollo de la presente propuesta tecnológica.

Asimismo, agradezco a mis compañeros de equipo por el trabajo realizado durante el desarrollo de InkFall, por la colaboración entre las diferentes áreas del proyecto y por las experiencias compartidas durante este proceso.

Finalmente, agradezco a todas las personas que contribuyeron directa o indirectamente a mi formación académica y a la culminación de esta etapa.

Resumen

La presente propuesta tecnológica tuvo como objetivo desarrollar y documentar el proceso de creación de assets tridimensionales para el videojuego InkFall, con énfasis en personajes, escenarios y objetos interactivos destinados a entornos de renderizado en tiempo real. El proyecto respondió a la necesidad de establecer un flujo de trabajo organizado que garantizara coherencia visual, eficiencia técnica e integración entre las áreas de arte, diseño y programación. Para ello, se aplicó un pipeline de producción 3D compuesto por planificación, modelado, escultura digital, retopología, mapeado UV, generación de mapas mediante baking, texturizado con materiales PBR, rigging, animación, exportación e integración en Unity. Blender, Substance Painter y Unity fueron las principales herramientas utilizadas durante el proceso. La validación permitió comprobar que los modelos cumplían con los criterios visuales y técnicos definidos para InkFall y que podían integrarse correctamente en el motor gráfico. También se identificaron aspectos de mejora relacionados con el nivel de detalle, la distribución de materiales y la optimización de determinadas geometrías. A partir de estas observaciones se realizaron ajustes en la topología, las texturas y la configuración de materiales, obteniendo assets con mayor consistencia visual y mejor adaptación al rendimiento en tiempo real. Se concluye que el pipeline aplicado fue adecuado para la producción e integración de recursos tridimensionales y proporciona una base técnica replicable para futuras etapas y proyectos de videojuegos.

Palabras Clave: Assets tridimensionales, modelado 3D, pipeline de producción, videojuegos, renderizado en tiempo real.

Abstract

This technological project aimed to develop and document the production workflow for three-dimensional assets used in the video game InkFall, with a focus on characters, environments, and interactive objects intended for real-time rendering. The project addressed the need for a structured workflow capable of ensuring visual consistency, technical efficiency, and effective integration across the art, design, and programming areas. A complete 3D production pipeline was implemented, including planning, modeling, digital sculpting, retopology, UV mapping, texture map baking, PBR-based texturing, rigging, animation, export, and integration into Unity. Blender, Substance Painter, and Unity were the main software tools employed throughout the process. The validation stage confirmed that the developed models met the visual and technical requirements established for InkFall and could be successfully integrated into the game engine. The evaluation also identified areas for improvement related to the level of detail, material distribution, and optimization of specific geometries. Based on these findings, adjustments were made to topology, textures, and material configurations, resulting in assets with greater visual consistency and improved suitability for real-time performance. The results indicate that the implemented pipeline provided an effective workflow for producing and integrating three-dimensional game assets and established a replicable technical foundation for future development stages and video game projects.

Keywords: *Three-dimensional assets, 3D modeling, production pipeline, video games, real-time rendering.*

Índice de contenido

Declaratoria de autoría y responsabilidad	II
Certificación.....	III
Dedicatoria.....	IV
Agradecimiento.....	V
Resumen	VI
Abstract.....	VII
Índice de contenido.....	VIII
Índice de tablas	XII
Índice de ilustraciones	XIII
Introducción.....	1
Capítulo I	4
1. Marco Referencial	4
1.1. Contextualización del Problema	4
1.2. Problemática.....	5
1.3. Formulación del Problema.....	6
1.4. Justificación de la investigación.....	6
1.5. Objetivo General.....	8
1.6. Objetivos Especificos	8
1.7. Alcance del proyecto	9
1.8. Delimitaciones	10
1.8.1. Delimitación geográfica.....	10
1.8.2. Delimitación temporal.....	10
1.8.3. Delimitación metodológica.....	11
1.8.4. Delimitación tecnológica	11
1.8.5. Delimitación del producto	12
1.9. Beneficiarios	13
1.9.1. Beneficiario directo.....	13
1.9.2. Beneficiarios indirectos	13
1.10. Metodología	13
1.11. Resultados esperados	15
1.12. Relación con el protocolo de investigación.....	17
Capítulo II.....	20
2. Marco teórico.....	20
2.1. Antecedentes de la investigación.....	20
2.2. Bases teóricas.....	24

2.3.	Bases metodológicas.....	25
2.4.	Bases tecnológicas	27
2.4.1.	TripoAI	27
2.4.2.	Blender.....	28
2.4.3.	Substance 3D Painter	28
2.4.4.	Mixamo	29
2.4.5.	Adobe Photoshop	29
2.4.6.	Unity	29
2.5.	Arquitecturas, modelos y estándares.....	30
2.5.1.	Arquitectura del flujo de producción de assets	30
2.5.2.	Modelo de producción de assets tridimensionales	31
2.5.3.	Modelo de renderizado Physically Based Rendering (PBR)	32
2.5.4.	Formato de intercambio FBX.....	33
2.5.5.	Optimización geométrica para aplicaciones en tiempo real.....	34
2.6.	Herramientas de desarrollo.....	36
2.6.1.	Blender.....	37
2.6.2.	Substance 3D Painter	37
2.6.3.	Mixamo	38
2.6.4.	Adobe Photoshop	38
2.6.5.	Unity	38
2.7.	Marco conceptual.....	39
2.7.1.	Asset 3D	39
2.7.2.	Modelado tridimensional (3D Modeling).....	39
2.7.3.	Escultura digital.....	40
2.7.4.	Retopología.....	41
2.7.5.	Mapeado UV (UV Mapping).....	41
2.7.6.	Texturizado	42
2.7.7.	Physically Based Rendering (PBR)	43
2.7.8.	Rigging.....	44
2.7.9.	Baking.....	45
2.7.10.	Motor gráfico (Game Engine).....	45
2.7.11.	Blender.....	46
2.7.12.	Unity	46
2.7.13.	Pipeline de producción de assets 3D.....	47
2.8.	Relación entre la teoría y la propuesta tecnológica	48
Capítulo III		50
3.	Manual del usuario	50
3.1.	Introducción	50
3.2.	Requisitos del sistema	50

3.3.	Instalación.....	51
3.3.1.	Descarga del juego	51
3.3.2.	Descompresión de los archivos	51
3.3.3.	Ejecución del juego	51
3.3.4.	Inicio del videojuego	51
3.4.	Interfaz del juego	51
3.4.1.	Menú principal.....	52
3.4.2.	Pantalla del juego.....	55
3.5.	Controles.....	56
3.6.	Como jugar	56
3.6.1.	Inicio	57
3.6.2.	Primer nivel.....	57
3.6.3.	Segundo nivel	57
3.6.4.	Sistema de dificultad	58
3.7.	Créditos.....	58
Capítulo IV.....		59
4.	Manual técnico de implementación.....	59
4.1.	Producción de assets 3D	59
4.2.	Pipeline de producción artística.....	60
4.2.1.	Planificación e investigación de los assets.....	60
4.2.2.	Diseño preliminar de escenarios (Blocking)	63
4.2.3.	Desarrollo de escenarios tridimensionales.....	66
4.2.4.	Desarrollo de personajes tridimensionales	69
4.2.5.	Escultura digital.....	70
4.2.6.	Retopología.....	75
4.2.7.	Mapeado UV (UV Mapping).....	78
4.2.8.	Desarrollo de objetos y props tridimensionales	79
4.3.	Materiales y texturas	81
4.3.1.	Generación de mapas de malla mediante Baking	81
4.3.2.	Texturizado de los assets tridimensionales	85
4.4.	Preparación y animación de personajes y objetos.....	90
4.4.1.	Rigging de personajes.....	90
4.4.2.	Animación de personajes	95
4.5.	Integración de recursos visuales en el motor	99
4.5.1.	Exportación de assets	99
4.5.2.	Integración de los assets en Unity.....	102
4.6.	Inventario de assets desarrollados.....	104
4.7.	Validación técnica de recursos visuales.....	105
4.7.1.	Criterios de validación de los assets	105

4.7.2. Resultados de la validación en Unity.....	107
4.8. Conclusiones	108
4.9. Recomendaciones	109
Referencias.....	111

Índice de tablas

Tabla 1	Correspondencia entre el protocolo de investigación y la propuesta tecnológica.....	19
Tabla 2	Entorno técnico utilizado durante la producción de assets de InkFall	36
Tabla 3	Requisitos mínimos del sistema	50
Tabla 4	Controles	56
Tabla 5	Integrantes del proyecto y su rol.....	58
Tabla 6	Mapas generados durante el proceso de baking.....	83
Tabla 7	Principales mapas de textura utilizados en el flujo PBR	90
Tabla 8	Inventario de assets desarrollados	105
Tabla 9	Criterios de validación de los recursos visuales	106
Tabla 10	Resultados de la validación técnica de los assets.....	107

Índice de ilustraciones

Ilustración 1	Flujo general de producción de assets tridimensionales	31
Ilustración 2	Menú principal	52
Ilustración 3	Pestaña de Chapters.	53
Ilustración 4	Pestaña de extras.	54
Ilustración 5	Pestaña de Settings.	55
Ilustración 6	Referencias visuales de los personajes de InkFall	62
Ilustración 7	Referencias visuales para el diseño de armas.....	63
Ilustración 8	Blocking del laberinto del nivel Concret Loop.	64
Ilustración 9	Blocking del escenario Crow, the Drowned Clown	65
Ilustración 10	Curva que define el recorrido inicial del laberinto.	66
Ilustración 11	Configuración del perfil utilizado para generar las paredes del laberinto	67
Ilustración 12	Conversión de la curva a malla (Convert to Mesh).	68
Ilustración 13	Despliegue UV del laberinto mediante Smart UV Project.....	68
Ilustración 14	Modelo base generado mediante inteligencia artificial	70
Ilustración 15	Proceso de escultura digital del personaje en Blender	72
Ilustración 16	Refinamiento de detalles mediante escultura digital.....	73
Ilustración 17	Modelo High Poly finalizado.	74
Ilustración 18	Proceso de retopología.	76
Ilustración 19	Modelo Low Poly finalizado.....	77
Ilustración 20	Resultado final del mapa UV.	79
Ilustración 21	Modelado 3D de trampas del nivel Concret Loop	80
Ilustración 22	Modelado de armas para el primer nivel de Concret Loop	81
Ilustración 23	Configuración para la generación de mapas de malla en Substance 3D Painter	82
Ilustración 24	Mapas de malla generados mediante baking en Substance 3D Painter	84
Ilustración 25	Opción de utilizar malla poligonal baja como malla poligonal alta	85
Ilustración 26	Texturización de modelo 3D.....	86
Ilustración 27	Aplicación de generadores de Dirt para el monstruo del laberinto	87
Ilustración 28	Texturización de armas del primer nivel.....	88
Ilustración 29	Texturización de trampas del primer nivel.....	88
Ilustración 30	Textura creada para el mapa del laberinto	89
Ilustración 31	Configuración del auto-rigging del personaje humanoide en Mixamo.	91
Ilustración 32	Personaje humanoide importado en Blender con el esqueleto generado por Mixamo.	92
Ilustración 33	Sistema de rigging manual del monstruo de seis patas.	93
Ilustración 34	Configuración del sistema de Cinemática Inversa (IK) en las patas del monstruo.	94
Ilustración 35	Proceso de Weight Paint.	95
Ilustración 36	Selección de animaciones para los personajes humanoides en Mixamo.	96
Ilustración 37	Creación manual de la animación del monstruo de seis patas en Blender.	97
Ilustración 38	Prueba de animación procedural del monstruo utilizando el asset de Unity.	98
Ilustración 39	Organización de animaciones en carpeta.	99
Ilustración 40	Estructura de organización de archivos de un asset 3D	100
Ilustración 41	Estructura de organización de archivos de un personaje humanoide	101
Ilustración 42	Integración de personajes dentro de Unity.	104

Introducción

El desarrollo de videojuegos integra diferentes áreas técnicas y artísticas que trabajan de manera conjunta para construir experiencias interactivas. Dentro de este proceso, la producción de recursos tridimensionales constituye una actividad fundamental para representar personajes, escenarios, enemigos y objetos que forman parte del entorno virtual. La creación de estos recursos involucra distintas etapas, entre ellas planificación, modelado tridimensional, escultura digital, retopología, mapeado UV, baking, texturizado, rigging, animación y preparación para su incorporación dentro de un motor gráfico.

En este contexto se desarrolla InkFall, un videojuego creado como proyecto colaborativo dentro del Trabajo de Integración Curricular de la carrera de Ingeniería en Realidad Virtual y Videojuegos. La producción artística del prototipo requirió desarrollar y preparar diferentes assets tridimensionales de acuerdo con los requerimientos visuales y técnicos establecidos para sus niveles. Debido a las características particulares de cada recurso, fue necesario aplicar diferentes procedimientos de producción y mantener una organización que permitiera relacionar las etapas de creación con su posterior incorporación dentro del motor gráfico Unity.

La presente propuesta tecnológica, titulada “Modelado, texturizado y animación de personajes y escenarios para el videojuego InkFall”, se enfoca en el desarrollo y documentación de los recursos tridimensionales correspondientes al área de producción 3D del proyecto. El trabajo comprende la planificación de los assets, el desarrollo de personajes y enemigos, el modelado tridimensional del laberinto correspondiente al nivel Concret Loop y la producción de diferentes objetos interactivos. Asimismo, se documentan los procesos de texturizado, rigging, animación, exportación, integración y comprobación técnica de los

recursos desarrollados. El diseño preliminar mediante blocking también fue utilizado como parte de la planificación de los niveles, mientras que el desarrollo tridimensional del escenario realizado dentro de esta propuesta se concentró específicamente en el laberinto de Concret Loop.

Para el desarrollo de la propuesta se empleó un flujo de producción organizado de manera secuencial y adaptado a las características de los diferentes tipos de assets. El proceso se estructuró mediante etapas de análisis y planificación, organización del flujo de trabajo, producción de recursos tridimensionales, rigging y animación, así como integración y validación técnica. Durante estas actividades se utilizaron herramientas como Blender para los principales procesos de producción tridimensional, Substance 3D Painter para baking y texturizado, Mixamo como apoyo para el rigging y animación de personajes humanoides, Tripo AI como herramienta de apoyo para la obtención de geometrías iniciales en determinados personajes y Unity para la integración y comprobación de los recursos dentro del prototipo. Estas herramientas forman parte del entorno tecnológico documentado a lo largo de la propuesta.

El documento se encuentra organizado en cuatro capítulos. El Capítulo I presenta el marco referencial de la propuesta, incluyendo la contextualización y formulación del problema, justificación, objetivos, alcance, delimitaciones, beneficiarios, metodología general y resultados esperados. El Capítulo II desarrolla el marco teórico mediante antecedentes, bases teóricas, metodológicas y tecnológicas, así como los principales conceptos, herramientas y fundamentos relacionados con la producción de assets tridimensionales. El Capítulo III corresponde al manual del usuario del videojuego InkFall, donde se describen los requisitos, instalación, interfaz, controles y funcionamiento general del prototipo. Finalmente, el Capítulo IV presenta el manual técnico de implementación, en

el cual se documenta el proceso de producción artística realizado, desde la planificación y desarrollo de los recursos tridimensionales hasta su exportación, integración y validación técnica dentro de Unity.

Capítulo I

1. Marco Referencial

1.1. Contextualización del Problema

En la actualidad, la industria de los videojuegos constituye uno de los sectores de mayor desarrollo dentro del entretenimiento digital, impulsado por los avances en motores gráficos, herramientas de modelado tridimensional y tecnologías de renderizado en tiempo real. Este crecimiento ha permitido desarrollar experiencias visuales cada vez más complejas, en las cuales los recursos gráficos desempeñan un papel importante en la representación de personajes, escenarios y elementos interactivos [1], [2].

Dentro del desarrollo de un videojuego, los assets tridimensionales constituyen recursos fundamentales para la construcción visual de los entornos y la representación de los elementos con los que interactúa el jugador. Su producción comprende diferentes etapas técnicas y artísticas, entre ellas planificación, modelado tridimensional, escultura digital, retopología, mapeado UV, texturizado, rigging, animación y preparación para su incorporación en un motor gráfico [3], [4].

La cantidad y variedad de procesos involucrados en la producción de assets 3D genera la necesidad de mantener criterios consistentes durante su desarrollo, debido a que las decisiones tomadas en una etapa pueden influir en las posteriores. Aspectos relacionados con la geometría, las coordenadas UV, los materiales, las estructuras de rigging, las animaciones y los formatos de exportación deben responder tanto a los requerimientos visuales del proyecto como a las condiciones técnicas necesarias para su posterior utilización dentro del motor gráfico [5], [6], [1].

Esta situación adquiere especial relevancia en proyectos académicos desarrollados de manera colaborativa, donde los recursos visuales deben integrarse con otras áreas como programación, diseño de niveles y mecánicas de juego. En el caso de InkFall, fue necesario producir personajes, enemigos, escenarios y objetos interactivos con características distintas para los niveles implementados en el prototipo. La diversidad de estos recursos implicó diferentes procedimientos de producción y preparación, generando la necesidad de mantener coherencia entre los requerimientos artísticos definidos para el videojuego y las características técnicas necesarias para su incorporación dentro de Unity [7], [1], [6].

1.2. Problemática

La producción de assets tridimensionales para videojuegos constituye un proceso que involucra diferentes etapas técnicas y artísticas, desde la planificación y el modelado hasta el texturizado, rigging, animación y preparación de los recursos para su utilización dentro de un motor gráfico. Debido a la dependencia existente entre estas etapas, las decisiones realizadas durante una fase pueden repercutir en los procedimientos posteriores y afectar las características finales del recurso desarrollado [4], [3].

Esta situación adquiere mayor complejidad cuando dentro de un mismo proyecto deben producirse recursos con características diferentes, como personajes, enemigos, escenarios, armas, trampas y objetos interactivos. Cada tipo de asset puede requerir procedimientos distintos de modelado, optimización geométrica, mapeado UV, texturizado, rigging o animación, lo que dificulta mantener criterios uniformes durante su preparación e integración.

En proyectos académicos de desarrollo de videojuegos, además, la documentación técnica del proceso de producción puede quedar relegada frente a la necesidad de completar el producto final. Esta situación dificulta el registro de los procedimientos aplicados, la

identificación de las características técnicas de los recursos y la comprensión de las etapas necesarias para preparar los assets antes de su incorporación al motor gráfico [8].

En el caso de InkFall, la producción artística requirió desarrollar y preparar diferentes recursos tridimensionales destinados a los niveles implementados en el prototipo. Las diferencias existentes entre personajes, enemigos, escenarios y objetos interactivos implicaron la utilización de distintos procedimientos y herramientas durante su producción. Como consecuencia, uno de los principales problemas dentro del área de producción 3D fue mantener una relación coherente entre los requerimientos visuales establecidos para cada recurso y las características técnicas necesarias para su posterior incorporación y comprobación dentro del motor gráfico Unity [5], [6], [4].

1.3. Formulación del Problema

A partir de la problemática identificada en el desarrollo de los recursos tridimensionales para el videojuego InkFall, se plantea la siguiente pregunta de investigación:

¿Cómo desarrollar el modelado, texturizado y animación de personajes y escenarios para el videojuego InkFall, considerando los requerimientos visuales y técnicos necesarios para su posterior integración y comprobación dentro del motor gráfico Unity?

1.4. Justificación de la investigación

La producción de recursos tridimensionales constituye una parte fundamental dentro del desarrollo del videojuego InkFall, debido a que personajes, enemigos, escenarios y objetos interactivos permiten representar visualmente los elementos definidos durante el diseño del proyecto. Estos recursos deben responder tanto a la identidad visual establecida como a las características técnicas necesarias para su incorporación dentro del motor gráfico Unity [4], [9], [10].

Desde el punto de vista tecnológico, el desarrollo de assets tridimensionales requiere la aplicación organizada de diferentes procedimientos relacionados con modelado, escultura digital, retopología, mapeado UV, baking, texturizado, rigging y animación. La aplicación de estos procesos permite preparar los recursos de acuerdo con sus características particulares y facilita su posterior incorporación y configuración dentro de un entorno de renderizado en tiempo real [3], [11].

La propuesta también se justifica por la necesidad de documentar el proceso de producción aplicado durante el desarrollo de los recursos visuales de InkFall. El registro de las etapas, herramientas y procedimientos empleados permite identificar cómo fueron preparados los diferentes assets y proporciona una referencia técnica sobre las decisiones adoptadas durante su desarrollo e integración en Unity. Esta documentación resulta especialmente relevante dentro de un proyecto colaborativo, donde los recursos producidos deben incorporarse posteriormente con otros componentes desarrollados por las distintas áreas del equipo.

En el ámbito académico, el desarrollo de la propuesta permite aplicar de manera práctica conocimientos relacionados con modelado tridimensional, producción de contenido digital, texturizado, animación y motores gráficos. Asimismo, la documentación del proceso realizado permite evidenciar la aplicación de los conocimientos técnicos adquiridos durante la formación en la carrera de Ingeniería en Realidad Virtual y Videojuegos [12], [13].

Por lo tanto, la importancia de la presente propuesta radica en desarrollar y documentar los recursos tridimensionales requeridos para el prototipo de InkFall mediante procedimientos técnicos acordes con las características de cada asset, permitiendo su posterior integración y comprobación dentro de Unity y generando, al mismo tiempo, una documentación del proceso de producción artística aplicado durante el proyecto.

1.5. Objetivo General

Desarrollar el modelado, texturizado y animación de personajes y escenarios para el videojuego InkFall mediante un flujo de trabajo estructurado para la producción de assets tridimensionales, considerando los requerimientos visuales y técnicos necesarios para su integración y comprobación dentro del motor gráfico Unity.

1.6. Objetivos Especificos

- Analizar metodologías, técnicas y herramientas relacionadas con el modelado, texturizado y animación de recursos tridimensionales para videojuegos, como fundamento técnico para el desarrollo artístico del proyecto.
- Estructurar un flujo de trabajo para la producción de los assets tridimensionales de InkFall, considerando las etapas de planificación, modelado, preparación geométrica, texturizado, rigging, animación, exportación e integración.
- Desarrollar los personajes, enemigos, escenario y objetos interactivos requeridos para el prototipo de InkFall mediante procesos de modelado tridimensional, escultura digital, retopología, mapeado UV, baking y texturizado, de acuerdo con las características de cada recurso.
- Implementar el rigging y la animación en los personajes y recursos que requieren movimiento, de acuerdo con sus características estructurales y las necesidades establecidas para el videojuego.
- Integrar los assets tridimensionales en el motor gráfico Unity, considerando la comprobación de su importación, escala, orientación, materiales, texturas, rigging y animaciones según las características de cada recurso.

1.7. Alcance del proyecto

La presente propuesta tecnológica comprende el desarrollo y documentación del proceso de producción de assets tridimensionales destinados al videojuego InkFall, considerando las actividades correspondientes al área de modelado 3D y producción artística realizadas durante el proyecto.

El alcance incluye la planificación de los recursos visuales a partir de las referencias y requerimientos establecidos por el equipo de desarrollo y en el Game Design Document (GDD); el diseño preliminar de los niveles mediante técnicas de blocking; el desarrollo tridimensional del laberinto correspondiente al nivel Concret Loop; y la producción de personajes, enemigos y objetos interactivos requeridos para los niveles implementados en el prototipo.

Dependiendo de las características de cada recurso, el proceso contempla etapas de modelado tridimensional, escultura digital, retopología, mapeado UV, generación de mapas mediante baking, texturizado, rigging y animación. Asimismo, se incluye la preparación de los assets para su utilización dentro del videojuego mediante su exportación en formato FBX, organización de texturas e integración en el motor gráfico Unity.

La propuesta se limita al área de producción de recursos tridimensionales y a las actividades directamente relacionadas con su preparación, integración y comprobación técnica dentro del prototipo. Por esta razón, no comprende el desarrollo de sistemas de programación, inteligencia artificial del videojuego, mecánicas de juego, sistemas multijugador, monetización, publicación comercial, estudios de experiencia de usuario ni pruebas de mercado.

De igual manera, el alcance se restringe a los recursos necesarios para los niveles implementados durante el Trabajo de Integración Curricular y no contempla la producción

de la totalidad de los contenidos proyectados en el Game Design Document (GDD) para futuras ampliaciones de InkFall.

1.8. Delimitaciones

La presente Propuesta Tecnológica se desarrolla bajo un conjunto de delimitaciones que establecen el contexto en el cual se ejecutó el proceso de producción de assets tridimensionales para el videojuego InkFall. Estas delimitaciones permiten precisar las condiciones geográficas, temporales, metodológicas y tecnológicas consideradas durante el desarrollo del proyecto.

1.8.1. Delimitación geográfica

El proyecto se desarrolló en el contexto académico de la Universidad Católica de Cuenca, dentro de la carrera de Ingeniería en Realidad Virtual y Videojuegos, como parte del Trabajo de Integración Curricular correspondiente al desarrollo del videojuego InkFall. Las actividades de planificación, producción de recursos tridimensionales, integración y documentación fueron realizadas en función de los requerimientos establecidos para el proyecto académico.

1.8.2. Delimitación temporal

El desarrollo de la propuesta se llevó a cabo durante el período correspondiente al Trabajo de Integración Curricular del año 2026. Durante este intervalo se ejecutaron las actividades de planificación, diseño preliminar, producción de assets tridimensionales, texturizado, rigging, animación, exportación, integración en el motor gráfico Unity y documentación técnica del proceso realizado.

La delimitación temporal comprende las actividades y recursos desarrollados durante dicho período para los niveles implementados en el prototipo de InkFall. Los contenidos

planteados para futuras ampliaciones del videojuego quedan fuera del período de ejecución de la presente propuesta.

1.8.3. Delimitación metodológica

La propuesta se delimitó metodológicamente al proceso de producción artística de assets tridimensionales para videojuegos, mediante la aplicación de un flujo de trabajo estructurado que organiza de manera secuencial las diferentes etapas necesarias para el desarrollo y preparación de los recursos visuales.

El flujo de trabajo aplicado comprende la revisión de referencias y requerimientos, planificación de los recursos, diseño preliminar mediante blocking cuando corresponde, modelado tridimensional, escultura digital, retopología, mapeado UV, generación de mapas mediante baking, texturizado, rigging, animación, exportación e integración dentro del motor gráfico Unity. La aplicación de cada etapa depende de las características particulares del asset, debido a que no todos los recursos requieren la totalidad de los procedimientos descritos.

La validación metodológica se limita a la comprobación técnica y visual de los recursos desarrollados una vez incorporados en Unity, considerando aspectos relacionados con la importación de la geometría, escala, orientación, materiales, texturas, rigging, animaciones e integración dentro de las escenas correspondientes. No se contemplan estudios de experiencia de usuario, pruebas de mercado ni evaluaciones generales de rendimiento del videojuego.

1.8.4. Delimitación tecnológica

El desarrollo de los recursos tridimensionales se realizó principalmente mediante Blender, utilizado para los procesos de modelado, escultura digital, retopología, mapeado UV y preparación de determinados sistemas de rigging y animación. Substance 3D Painter

fue empleado para la generación de mapas mediante baking y el texturizado de los modelos, mientras que Unity se utilizó como motor gráfico para la integración y comprobación de los recursos dentro del videojuego.

Adobe Photoshop fue utilizado durante la etapa de diseño preliminar para elaborar representaciones de la distribución de determinados escenarios. De manera complementaria, Mixamo fue empleado en personajes humanoides para procesos de rigging y obtención de animaciones base. Las herramientas utilizadas se limitaron a aquellas necesarias para desarrollar el flujo de producción artística aplicado durante el proyecto.

De manera complementaria, se utilizó TripoAI durante la etapa inicial de producción de determinados personajes. Esta herramienta permitió obtener modelos tridimensionales base que fueron utilizados únicamente como punto de partida para el desarrollo. Posteriormente, estos recursos fueron importados en Blender y sometidos a procesos de corrección, escultura digital, retopología, mapeado UV, baking, texturizado, rigging y animación antes de su incorporación al videojuego.

1.8.5. Delimitación del producto

Como resultado de la investigación se obtendrá un conjunto de personajes, enemigos, el escenario tridimensional correspondiente al nivel Concret Loop y objetos interactivos preparados para su utilización dentro del videojuego InkFall. Asimismo, se contará con las representaciones preliminares elaboradas mediante blocking durante la planificación de los niveles del prototipo.

No se contempla el lanzamiento comercial del videojuego ni el desarrollo de todos los niveles proyectados en el Game Design Document, sino únicamente aquellos definidos dentro del alcance del Trabajo de Integración Curricular.

1.9. Beneficiarios

1.9.1. Beneficiario directo

Los beneficiarios directos de la propuesta son los integrantes del equipo de desarrollo del videojuego InkFall, debido a que disponen de los recursos tridimensionales desarrollados y preparados para su incorporación dentro del prototipo. Asimismo, la documentación del proceso de producción permite identificar las etapas, herramientas y procedimientos aplicados a los diferentes assets, facilitando su comprensión, integración y posterior modificación dentro del proyecto.

1.9.2. Beneficiarios indirectos

Como beneficiarios indirectos se consideran los estudiantes y docentes de la carrera de Ingeniería en Realidad Virtual y Videojuegos de la Universidad Católica de Cuenca, quienes pueden utilizar la documentación desarrollada como referencia académica para comprender las diferentes etapas involucradas en la producción de assets tridimensionales destinados a videojuegos.

De igual manera, la documentación técnica puede servir como material de consulta para futuros proyectos académicos relacionados con modelado 3D, producción de personajes y escenarios, texturizado, animación e integración de recursos gráficos en motores de videojuegos, proporcionando una referencia práctica sobre el flujo de trabajo aplicado durante el desarrollo de InkFall.

1.10. Metodología

La presente propuesta tecnológica se desarrolló mediante una metodología de investigación aplicada, con alcance descriptivo y apoyo de investigación documental. El carácter aplicado se relaciona con la utilización de conocimientos y procedimientos técnicos para el desarrollo de los recursos tridimensionales requeridos por el videojuego InkFall. El

alcance descriptivo permitió registrar de manera sistemática las etapas, herramientas y procedimientos empleados durante la producción de los assets, mientras que la investigación documental se utilizó para fundamentar las técnicas y tecnologías empleadas mediante la revisión de literatura especializada, artículos científicos, manuales técnicos y documentación oficial [14].

Para el desarrollo tecnológico se empleó un flujo de producción organizado de manera secuencial, basado en el modelo en cascada (Waterfall) [8] y adaptado a las características del pipeline de producción de assets tridimensionales. Las fases presentan una dependencia técnica, debido a que los resultados obtenidos durante determinados procesos, como el esculpido, la retopología, el mapeado UV o el texturizado, constituyen la base para las actividades posteriores. No obstante, dentro de cada fase se realizaron revisiones y ajustes cuando las características de los recursos lo requirieron antes de continuar con las siguientes etapas.

La metodología se organizó en cinco fases relacionadas con los objetivos específicos de la propuesta:

Fase 1. Análisis y planificación. Se revisaron metodologías, técnicas, herramientas, referencias visuales y los requerimientos establecidos para el proyecto, incluyendo la información contenida en el Game Design Document (GDD). Esta fase permitió identificar las características visuales y técnicas de los recursos que debían desarrollarse y establecer las referencias necesarias para orientar su producción.

Fase 2. Estructuración del flujo de trabajo. A partir de los requerimientos identificados se organizó el pipeline de producción de los assets tridimensionales, considerando las etapas de planificación, modelado, preparación geométrica, texturizado,

rigging, animación, exportación e integración. Debido a las diferencias existentes entre los tipos de recursos, no todos los assets requirieron la totalidad de las etapas definidas.

Fase 3. Producción de assets tridimensionales. Se desarrollaron los recursos visuales requeridos para el prototipo mediante los procedimientos correspondientes a las características de cada asset. El proceso comprendió, según el tipo de recurso, etapas de modelado tridimensional, escultura digital, retopología, mapeado UV, generación de mapas mediante baking y texturizado. Para determinados personajes, Tripo AI se utilizó únicamente como herramienta de apoyo para obtener una geometría inicial, la cual posteriormente fue modificada y preparada mediante las etapas correspondientes del flujo de producción.

Fase 4. Rigging y animación. Los personajes y recursos que requerían movimiento fueron preparados mediante sistemas de rigging y animaciones acordes con sus características estructurales. Para los personajes humanoides se utilizaron procedimientos asistidos mediante Mixamo, mientras que los recursos con estructuras diferentes requirieron sistemas desarrollados directamente en Blender.

Fase 5. Integración y validación técnica. Los assets desarrollados fueron exportados y posteriormente incorporados al proyecto de Unity. La validación se realizó mediante comprobaciones técnicas y visuales relacionadas con la importación de la geometría, escala, orientación, materiales, texturas, rigging, animaciones y utilización de los recursos dentro de las escenas correspondientes. La evaluación se limitó a los recursos tridimensionales desarrollados o preparados dentro de la propuesta y no comprendió pruebas de experiencia de usuario ni una evaluación integral del rendimiento del videojuego.

1.11. Resultados esperados

Como resultado de la presente Propuesta Tecnológica se espera disponer de un conjunto de assets tridimensionales desarrollados e integrados correctamente en el

videojuego InkFall, cumpliendo con los requerimientos técnicos y artísticos definidos durante el proyecto.

Entre los principales resultados esperados se encuentran:

- La identificación de metodologías, técnicas y herramientas relacionadas con el modelado, texturizado y animación de recursos tridimensionales para videojuegos, como fundamento técnico para el desarrollo artístico del proyecto.
- La estructuración de un flujo de trabajo para organizar las diferentes etapas de producción de los assets tridimensionales de InkFall.
- El desarrollo de personajes, enemigos, el escenario tridimensional correspondiente al laberinto de Concret Loop y diferentes objetos interactivos mediante los procedimientos de modelado tridimensional, escultura digital, retopología, mapeado UV, baking y texturizado aplicables a cada recurso.
- La implementación de sistemas de rigging y animaciones en los personajes y recursos que requieren movimiento, de acuerdo con sus características estructurales y las necesidades establecidas para el videojuego.
- La integración de los assets tridimensionales dentro del motor gráfico Unity y la comprobación de aspectos relacionados con su importación, escala, orientación, materiales, texturas, rigging y animaciones, según corresponda a cada recurso.

En conjunto, estos resultados permitirán evidenciar el cumplimiento de los objetivos planteados mediante el desarrollo, preparación e integración de los recursos tridimensionales requeridos para el prototipo de InkFall.

1.12. Relación con el protocolo de investigación

La presente propuesta tecnológica mantiene correspondencia con el anteproyecto elaborado inicialmente para el Trabajo de Integración Curricular, denominado “Modelado, Texturizado y Animación de Personajes y Escenarios para el Videojuego InkFall”. El planteamiento inicial estableció como eje principal la producción de recursos tridimensionales para el videojuego mediante procesos de modelado, texturizado, animación y posterior integración dentro del motor gráfico Unity.

El objetivo general planteado inicialmente propuso desarrollar el modelado, texturizado y animación de personajes y escenarios para InkFall mediante Blender y su integración en Unity. Durante el desarrollo de la propuesta tecnológica este propósito se mantuvo, aunque su redacción fue ajustada para incorporar un flujo de trabajo estructurado y delimitar la comprobación de los recursos a aspectos técnicos relacionados con su integración dentro del motor gráfico.

De igual manera, los objetivos específicos del anteproyecto contemplaron actividades relacionadas con el análisis de referencias visuales y técnicas, el diseño preliminar de los niveles, la creación de modelos tridimensionales, el texturizado, el rigging, la animación y la integración de los recursos. En la propuesta tecnológica estas actividades fueron reorganizadas para establecer una correspondencia más clara entre el análisis técnico, la estructuración del flujo de trabajo, la producción de los assets tridimensionales, el rigging y la animación, y finalmente su integración y comprobación dentro de Unity.

El alcance inicialmente planteado consideró la producción de personajes, escenarios y elementos interactivos, además de su posterior integración dentro del motor Unity. Durante el desarrollo del proyecto, este alcance fue adaptado a las necesidades reales del prototipo y

a los recursos efectivamente requeridos por los niveles implementados, manteniendo como área principal de trabajo la producción de assets tridimensionales.

En cuanto a la metodología, el anteproyecto estableció un proceso organizado por etapas que comprendía análisis de referencias, diseño preliminar de escenarios, modelado, retopología, mapeado UV, texturizado, rigging, animación, exportación e integración en Unity. La presente propuesta tecnológica conserva esta lógica general mediante un flujo de producción secuencial, incorporando revisiones y ajustes dentro de cada fase cuando las características de los recursos lo requirieron.

Por lo tanto, existe continuidad entre el planteamiento inicial y la propuesta desarrollada, manteniéndose el área de especialización, el objeto de estudio y las principales actividades relacionadas con la producción tridimensional. Los ajustes realizados corresponden principalmente a la delimitación del alcance y a una mayor precisión en la organización metodológica y en los criterios utilizados para comprobar técnicamente los recursos desarrollados.

Tabla 1

Correspondencia entre el protocolo de investigación y la propuesta tecnológica

Elemento	Protocolo de investigación	Propuesta Tecnológica
Tema	Modelado, texturizado y animación de personajes y escenarios para InkFall.	Se mantiene el mismo tema y área de especialización.
Propósito	Desarrollar assets 3D mediante Blender e integrarlos en Unity.	Desarrollar assets mediante un flujo estructurado y prepararlos para su integración y comprobación técnica en Unity.
Referencias y planificación	Análisis del GDD y referencias visuales y técnicas.	Se incorpora el análisis de requerimientos y referencias como primera fase del flujo de trabajo.
Diseño de escenarios	Desarrollo de blocking para definir distribución y proporciones.	Se mantiene el blocking como actividad de planificación de los niveles, mientras que el desarrollo tridimensional del escenario corresponde al laberinto de Concret Loop.
Producción 3D	Modelado de personajes y escenarios considerando topología y nivel de detalle.	Se incorporan procesos de modelado tridimensional, escultura digital, retopología y preparación de la geometría según las características de cada asset.
Texturizado	Aplicación de materiales y mapas UV.	Se incorporan mapeado UV, baking y texturizado mediante Substance 3D Painter.
Rigging y animación	Implementación de sistemas de rigging y animaciones funcionales.	Se aplican procesos de rigging y animación según las características y necesidades de cada personaje o recurso.
Integración	Importación de los recursos dentro de Unity.	Se incorporan la exportación en formato FBX, configuración de materiales y texturas, integración y comprobación técnica de los recursos dentro de Unity.
Alcance	Producción de los recursos visuales inicialmente previstos para el videojuego.	El alcance se adapta a los assets efectivamente requeridos por los niveles implementados en el prototipo.

Nota. Elaboración propia a partir del anteproyecto y del desarrollo de la Propuesta Tecnológica.

Capítulo II

2. Marco teórico

2.1. Antecedentes de la investigación

La producción de recursos tridimensionales destinados a videojuegos ha sido abordada en diferentes investigaciones relacionadas con la organización de pipelines de producción, modelado tridimensional, preparación de geometría, texturizado, integración con motores gráficos y, recientemente, utilización de herramientas de inteligencia artificial durante determinadas etapas del desarrollo. La revisión de estos trabajos permite identificar procedimientos y criterios técnicos aplicables a la creación de assets tridimensionales y establecer su relación con el flujo de producción empleado durante el desarrollo del videojuego InkFall.

Zavgorodnyaya y Zelenova [15] analizaron un pipeline sistemático orientado a la creación de assets tridimensionales destinados a proyectos de videojuegos. El proceso descrito por las autoras comprende diferentes etapas de producción, entre ellas el desarrollo de modelos Low Poly y High Poly, la organización de la geometría, el mapeado UV, la exportación mediante formato FBX y la generación de mapas de detalle mediante baking en Substance 3D Painter. Asimismo, el estudio destaca la importancia de mantener criterios consistentes de organización, nomenclatura y transferencia de recursos entre las diferentes herramientas utilizadas durante el desarrollo. Este antecedente se relaciona con la presente propuesta tecnológica debido a que el proceso aplicado en InkFall también se estructura mediante diferentes etapas de producción, incluyendo modelado, retopología, mapeado UV, baking, texturizado y posterior preparación de los recursos para su integración en Unity.

Lee [16] analizó el uso de Blender 3D como herramienta para el desarrollo de videojuegos independientes, considerando las dificultades económicas y técnicas que pueden surgir por la dependencia de software comercial especializado. El estudio examina la aplicación de Blender dentro del flujo de producción y su interoperabilidad con motores gráficos como Unity y Unreal Engine, considerando su participación en diferentes etapas del desarrollo de contenido tridimensional. Asimismo, se analiza cómo la integración entre estas herramientas puede facilitar la transferencia de recursos desde su creación hasta su incorporación dentro del entorno de desarrollo del videojuego. Este antecedente se relaciona con la presente propuesta tecnológica debido a que Blender constituye la herramienta principal utilizada para la producción de los assets tridimensionales de InkFall, mientras que Unity se emplea para su posterior integración y comprobación dentro del prototipo.

Wu et al. [17] realizaron una revisión de los avances relacionados con la generación de recursos tridimensionales orientados a entornos interactivos, centrando su análisis en las características necesarias para que un asset pueda considerarse preparado para producción. Los autores organizan el proceso considerando aspectos como la generación de geometría, optimización de topología, mapeado UV, materiales PBR, rigging y preparación de escenas, señalando que la calidad visual de un modelo no constituye por sí sola un criterio suficiente para su utilización directa dentro de un motor gráfico. Asimismo, el estudio identifica una brecha entre los modelos tridimensionales generados automáticamente y los requerimientos técnicos necesarios para su implementación en aplicaciones interactivas. Este antecedente se relaciona con la presente propuesta debido a que el flujo aplicado en InkFall contempla diferentes etapas de preparación técnica antes de la integración de los assets en Unity, incluyendo retopología, mapeado UV, texturizado, rigging y animación según las características de cada recurso.

Parab et al. [18] desarrollaron una experiencia tridimensional utilizando Unity como motor gráfico para la construcción del entorno interactivo, incorporando recursos tridimensionales elaborados mediante Blender dentro del proceso de desarrollo. El trabajo evidencia la utilización conjunta de herramientas de producción 3D y motores gráficos, donde los modelos y escenarios creados previamente deben ser incorporados y configurados dentro del entorno de ejecución del videojuego. Este antecedente se relaciona con la presente propuesta tecnológica debido a que el flujo aplicado en InkFall también utiliza Blender para la producción y preparación de los recursos tridimensionales y Unity para su posterior integración, configuración y comprobación dentro de las escenas del prototipo.

Begemann y Hutson [19] evaluaron la utilización de herramientas de inteligencia artificial para la generación de assets tridimensionales destinados a videojuegos mediante la comparación de un flujo de producción desarrollado en Blender y un flujo asistido por herramientas como Meshy 6 y Hunyuan 3D. El estudio analizó aspectos relacionados con el tiempo de producción, la topología de los modelos, la organización de los mapas UV y el trabajo adicional requerido después de la generación automática. Los resultados mostraron que las herramientas de inteligencia artificial pueden reducir el tiempo necesario para obtener una primera versión del recurso; sin embargo, los modelos generados presentaron dificultades relacionadas con geometría densa, mapas UV fragmentados, edición de materiales y preparación técnica. Los autores concluyen que estas herramientas resultan especialmente útiles durante etapas de ideación y prototipado, mientras que la intervención del artista continúa siendo necesaria para procesos como retopología, optimización, reconstrucción UV, refinamiento de materiales y validación dentro del motor gráfico.

Este antecedente presenta una relación directa con el proceso desarrollado para determinados personajes de InkFall, debido a que los modelos generados mediante

inteligencia artificial fueron utilizados únicamente como una geometría inicial. Posteriormente, estos recursos fueron modificados en Blender y continuaron por etapas de escultura, retopología, mapeado UV, baking, texturizado, rigging y animación antes de su utilización dentro del prototipo.

En conjunto, los antecedentes revisados permiten identificar diferentes aspectos relevantes para la producción de recursos tridimensionales destinados a videojuegos. Los trabajos relacionados con pipelines de producción destacan la importancia de organizar el desarrollo de los assets mediante etapas como modelado, retopología, mapeado UV, baking y preparación para motores gráficos [15], [17]. Por otra parte, los estudios relacionados con Blender y Unity muestran la necesidad de mantener un flujo de trabajo organizado entre las herramientas utilizadas para la creación de los recursos y el motor gráfico donde posteriormente serán incorporados [16], [18].

De manera complementaria, las investigaciones relacionadas con la generación tridimensional mediante inteligencia artificial evidencian que estas tecnologías pueden utilizarse como herramientas de apoyo durante las etapas iniciales de producción, aunque los modelos obtenidos requieren intervención posterior para adecuar su geometría, topología, mapeado UV, materiales y otras características necesarias para su utilización en entornos interactivos [17], [19].

Estos antecedentes guardan relación con la presente propuesta tecnológica debido a que el desarrollo de los recursos tridimensionales de InkFall se realizó mediante un flujo estructurado que comprende diferentes etapas de producción y preparación antes de su incorporación en Unity. Asimismo, el proyecto integra el uso de Blender como herramienta principal de producción tridimensional, Substance 3D Painter para baking y texturizado, Mixamo para determinados procesos de rigging y animación, y Unity para la integración y

comprobación de los recursos dentro del prototipo. En el caso de los personajes generados inicialmente mediante inteligencia artificial, estos modelos fueron utilizados únicamente como punto de partida y posteriormente modificados mediante las etapas correspondientes del pipeline antes de su utilización dentro del videojuego.

2.2. Bases teóricas

La propuesta tecnológica se fundamenta en los principios de la producción de assets tridimensionales para videojuegos, entendida como el proceso de diseño, desarrollo y optimización de recursos gráficos destinados a ser utilizados en motores de renderizado en tiempo real. Esta producción comprende un conjunto de etapas interrelacionadas, entre ellas el modelado tridimensional, la escultura digital, la retopología, el mapeado UV, el texturizado, el rigging y la integración dentro del motor gráfico, las cuales buscan equilibrar la calidad visual con el rendimiento del videojuego. La correcta aplicación de este flujo de trabajo permite obtener modelos optimizados que cumplen con los requerimientos técnicos y artísticos del proyecto [1], [3].

Otra base teórica corresponde al concepto de pipeline de producción de assets, ampliamente utilizado en la industria del desarrollo de videojuegos. Un pipeline establece una secuencia organizada de procesos para garantizar la consistencia en la creación de recursos digitales, facilitar la colaboración entre diferentes áreas del proyecto y reducir errores durante la integración de los modelos en el motor gráfico. La implementación de un pipeline estructurado favorece la reutilización de recursos, mejora la trazabilidad del trabajo y optimiza los tiempos de producción [1], [4].

Asimismo, la propuesta se sustenta en los principios del renderizado basado en propiedades físicas (Physically Based Rendering - PBR). Este enfoque permite representar

las propiedades visuales de los materiales mediante parámetros y mapas de textura como Base Color, Normal, Roughness, Metallic y Ambient Occlusion. La utilización de materiales PBR permite organizar de manera consistente las características de las superficies dentro de entornos de renderizado en tiempo real. Sin embargo, su apariencia final puede variar según factores como los shaders, la iluminación, el render pipeline y la configuración utilizada dentro de cada motor gráfico [11], [2], [6].

Otro fundamento importante es la optimización de modelos para aplicaciones en tiempo real. A diferencia de los modelos destinados a animación cinematográfica, los videojuegos requieren recursos con una cantidad controlada de polígonos y texturas para mantener un rendimiento adecuado durante la ejecución. Técnicas como la retopología, el despliegue eficiente de coordenadas UV y la generación de mapas mediante baking permiten reducir la complejidad geométrica sin afectar significativamente la calidad visual del modelo, favoreciendo un uso eficiente de los recursos del sistema [20], [2], [1].

Finalmente, la propuesta se apoya en los principios del desarrollo colaborativo de videojuegos, donde la producción artística constituye una de las áreas que interactúa directamente con programación, diseño de niveles y narrativa. En este contexto, la documentación del proceso de creación de assets tridimensionales facilita la integración de los recursos dentro del proyecto, mejora la comunicación entre los integrantes del equipo y promueve la aplicación de buenas prácticas para el desarrollo de videojuegos [7], [21].

2.3. Bases metodológicas

La presente Propuesta Tecnológica se fundamenta en metodologías de investigación y de desarrollo que permitieron planificar, ejecutar y documentar de manera organizada el proceso de creación de los assets tridimensionales utilizados en el videojuego InkFall.

En cuanto a la metodología de investigación, el proyecto se desarrolló bajo un enfoque de investigación aplicada, ya que estuvo orientado a resolver un problema específico mediante el desarrollo de una solución tecnológica. Este tipo de investigación busca aplicar conocimientos científicos y técnicos para generar productos o procesos que respondan a necesidades concretas, en este caso, la creación e integración de assets tridimensionales optimizados para un videojuego desarrollado en Unity [14].

Asimismo, la investigación presenta un alcance descriptivo, debido a que permitió documentar de forma sistemática cada una de las etapas involucradas en el proceso de producción de los modelos tridimensionales. A través de este enfoque fue posible registrar las actividades realizadas, las herramientas empleadas y los procedimientos técnicos aplicados durante el modelado, texturizado, optimización e integración de los assets, generando una documentación técnica que puede servir como referencia para futuros proyectos de desarrollo de videojuegos [14].

Complementariamente, se recurrió a la investigación documental, sustentada en la revisión de libros especializados, artículos científicos, manuales técnicos y documentación oficial sobre modelado tridimensional, motores gráficos y desarrollo de videojuegos. La consulta de estas fuentes permitió fundamentar las decisiones técnicas adoptadas durante el desarrollo del proyecto y respaldar las metodologías implementadas en cada etapa de producción [14].

Respecto a la metodología de desarrollo, se empleó un pipeline de producción de assets tridimensionales para organizar de manera secuencial las diferentes etapas involucradas en la creación y preparación de los recursos gráficos. Este flujo comprende procesos como recopilación de referencias, modelado tridimensional, escultura digital

cuando corresponde, retopología, mapeado UV, baking, texturizado, rigging, animación, exportación e integración dentro del motor gráfico [1], [3].

La organización del pipeline mantuvo una secuencia general entre las diferentes etapas, debido a que los resultados obtenidos en determinados procesos constituyen la base para continuar con los procedimientos posteriores. No obstante, durante cada fase se realizaron revisiones y ajustes cuando las características del recurso lo requirieron antes de continuar con las siguientes etapas. Asimismo, no todos los assets atravesaron la totalidad de los procedimientos, ya que el flujo fue adaptado según las características técnicas y artísticas de cada recurso [8].

La combinación de una metodología de investigación aplicada, con alcance descriptivo y apoyo documental, junto con un pipeline de producción estructurado, permitió organizar y registrar el proceso de desarrollo de los assets tridimensionales de InkFall desde su planificación inicial hasta su integración y comprobación dentro del motor gráfico Unity [1], [3].

2.4. Bases tecnológicas

2.4.1. TripoAI

Tripo AI fue utilizada como herramienta de apoyo durante la etapa inicial de producción de determinados personajes de InkFall. Su función dentro del proyecto consistió en generar modelos tridimensionales base tomando como referencia los diseños previamente establecidos para los personajes.

Los modelos obtenidos mediante TripoAI no fueron utilizados directamente como assets finales, debido a que presentaban aspectos geométricos que requerían intervención posterior. Por esta razón, fueron importados en Blender, donde se realizaron correcciones de

forma y proporción antes de continuar con las etapas de escultura digital, retopología, mapeado UV, baking, texturizado, rigging y animación.

De esta manera, TripoAI se empleó únicamente como herramienta de apoyo para obtener una geometría inicial, mientras que la preparación técnica y artística necesaria para convertir estos modelos en recursos utilizables dentro del videojuego se desarrolló mediante las demás herramientas del pipeline.

2.4.2. Blender

Blender constituyó una de las principales herramientas empleadas durante el desarrollo del proyecto. Este software fue utilizado en diferentes etapas de la producción tridimensional, entre ellas el modelado, la escultura digital, la retopología, el mapeado UV, el rigging, la animación y la preparación de los recursos para su exportación. Su utilización permitió trabajar con personajes, enemigos, escenarios, armas, trampas y otros objetos tridimensionales requeridos para el desarrollo de InkFall [4], [5].

2.4.3. Substance 3D Painter

Para el proceso de materiales y texturas se utilizó Substance 3D Painter, herramienta empleada para realizar procedimientos de baking y desarrollar las texturas correspondientes a los distintos assets. Durante este proceso se generaron y utilizaron mapas relacionados con la representación de las características visuales de las superficies, tales como Base Color, Normal, Roughness y otros mapas complementarios requeridos durante el flujo de producción [11], [2], [6].

El proceso de texturizado se relacionó con los principios del Physically Based Rendering (PBR), utilizados para representar distintas propiedades visuales de los materiales mediante mapas de textura. La apariencia final de estos materiales depende de factores como

la interpretación de los canales, los shaders, la iluminación y la configuración del render pipeline utilizado dentro del motor gráfico. Por esta razón, los materiales y texturas desarrollados fueron posteriormente revisados y configurados dentro de Unity para adecuarlos a las características visuales requeridas por InkFall [11], [2], [6].

2.4.4. Mixamo

Como herramienta complementaria para la preparación de personajes humanoides se utilizó Mixamo, principalmente para procesos de rigging automático y para la obtención de determinadas animaciones base. Su utilización se aplicó a personajes compatibles con estructuras humanoides, como el protagonista y determinados enemigos. En recursos que presentaban características estructurales diferentes, el proceso de rigging se desarrolló directamente en Blender mediante la creación y configuración manual de sistemas de huesos.

2.4.5. Adobe Photoshop

Durante las etapas iniciales de planificación también se utilizó Adobe Photoshop como herramienta de apoyo para la elaboración de representaciones preliminares de determinados escenarios. Su utilización permitió plantear la distribución espacial y realizar propuestas de blocking antes de continuar con la construcción tridimensional de los niveles en Blender.

2.4.6. Unity

La integración de los recursos tridimensionales se realizó mediante Unity, motor gráfico utilizado para incorporar los assets dentro del videojuego InkFall. En esta etapa se efectuaron procesos relacionados con la importación de los modelos, asignación y configuración de materiales y texturas, incorporación de animaciones y preparación de determinados elementos para su utilización dentro de las escenas del proyecto. Unity permitió

además visualizar los recursos en el contexto del videojuego y comprobar aspectos relacionados con su correcta incorporación y funcionamiento. [6], [1].

Para el intercambio de los modelos tridimensionales entre Blender y Unity se utilizó principalmente el formato FBX. Este formato permitió transferir información relacionada con la geometría, jerarquías, estructuras de huesos y animaciones de los recursos cuando correspondía. Sin embargo, los materiales y texturas requieren una comprobación y configuración posterior dentro del motor gráfico, debido a que su apariencia y funcionamiento no dependen únicamente de la información contenida en el archivo FBX. [1], [5].

2.5. Arquitecturas, modelos y estándares

2.5.1. Arquitectura del flujo de producción de assets

La producción de assets tridimensionales para videojuegos puede organizarse mediante un pipeline que establece una secuencia de etapas relacionadas entre sí. Este tipo de estructura permite ordenar el proceso desde la definición inicial de los requerimientos visuales hasta la preparación de los recursos para su incorporación en un motor gráfico.

De manera general, un pipeline de producción 3D puede comprender etapas de recopilación de referencias, modelado tridimensional, escultura digital cuando el recurso lo requiere, retopología, mapeado UV, baking, texturizado, rigging, animación, exportación e integración dentro del motor gráfico. La participación de cada etapa depende de las características y finalidad del recurso, debido a que un objeto estático no requiere necesariamente los mismos procedimientos que un personaje destinado a ser animado.

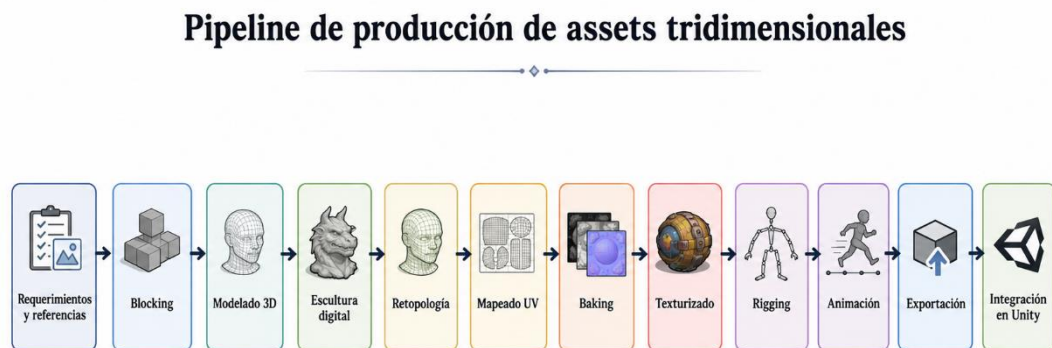
La relación entre las etapas del pipeline presenta dependencias técnicas. Por ejemplo, la geometría obtenida durante el modelado constituye la base para los procesos posteriores de preparación de topología y mapeado UV, mientras que los recursos destinados a la

animación requieren una estructura geométrica adecuada antes de incorporar sistemas de rigging. De esta manera, la organización del flujo permite identificar el orden y las relaciones existentes entre los distintos procedimientos involucrados en la producción tridimensional.

Asimismo, el pipeline puede adaptarse de acuerdo con las necesidades del proyecto y del tipo de asset desarrollado. Algunos recursos pueden producirse directamente mediante geometría de baja complejidad, mientras que otros requieren un proceso High Poly–Low Poly, transferencia de detalles mediante baking o etapas adicionales de rigging y animación. Por esta razón, el flujo de producción no implica que todos los recursos deban atravesar exactamente los mismos procedimientos, sino que establece una estructura general que puede ajustarse según las características técnicas y artísticas de cada asset [1], [3].

Ilustración 1

Flujo general de producción de assets tridimensionales



Nota. Representación general de las etapas que pueden conformar un pipeline de producción de assets tridimensionales.

Elaboración propia.

2.5.2. Modelo de producción de assets tridimensionales

La producción de assets tridimensionales puede organizarse mediante un modelo secuencial en el que las distintas etapas presentan una dependencia técnica entre sí. Bajo este enfoque, los resultados obtenidos en una fase constituyen la base para continuar con los

procedimientos posteriores, por lo que actividades como modelado, retopología, mapeado UV, texturizado, rigging y animación mantienen una relación de continuidad dentro del pipeline de producción [1], [3].

Este tipo de organización guarda relación con el modelo en cascada, caracterizado por estructurar el desarrollo mediante fases consecutivas. Su aplicación permite establecer un orden general de trabajo y definir los resultados requeridos antes de avanzar hacia las siguientes etapas del proceso.

Sin embargo, en la producción de recursos tridimensionales pueden presentarse revisiones y ajustes durante el desarrollo. La detección de problemas relacionados con la geometría, topología, coordenadas UV, materiales o deformación de una malla puede requerir modificaciones antes de continuar con las etapas posteriores. Por esta razón, aunque el pipeline mantenga una estructura general secuencial, puede incorporar ciclos de revisión destinados a corregir aspectos específicos del recurso antes de su finalización.

De esta manera, la organización secuencial y las revisiones durante el proceso no deben considerarse enfoques excluyentes. El modelo secuencial establece el orden general de las etapas de producción, mientras que las revisiones permiten realizar los ajustes necesarios para que cada recurso se encuentre preparado antes de continuar dentro del pipeline.

2.5.3. Modelo de renderizado Physically Based Rendering (PBR)

El Physically Based Rendering (PBR) corresponde a un enfoque utilizado para representar las propiedades visuales de los materiales mediante parámetros y mapas de textura que describen el comportamiento de las superficies frente a la iluminación. Dentro de este flujo pueden emplearse mapas como Base Color, Normal, Roughness, Metallic y

Ambient Occlusion, entre otros, de acuerdo con las características del material y del sistema de renderizado utilizado [11], [2], [6].

El mapa Base Color contiene la información cromática principal de la superficie, mientras que Roughness permite representar variaciones relacionadas con el grado de rugosidad del material. El mapa Metallic permite diferenciar el comportamiento de superficies metálicas y no metálicas, y el Normal Map se utiliza para representar detalles superficiales mediante información de normales sin necesidad de incorporar toda esa información directamente en la geometría del modelo.

La apariencia final de un material PBR depende también de la configuración empleada dentro de cada aplicación o motor gráfico. Factores como los shaders, la iluminación, el render pipeline, la interpretación de los canales de textura y la configuración de los materiales pueden producir diferencias visuales entre distintas herramientas [11], [2], [6].

Por esta razón, el flujo PBR permite organizar las propiedades visuales de los materiales mediante mapas y parámetros específicos, pero la representación final de cada superficie debe considerarse en función de las características del entorno de renderizado en el que será utilizada.

2.5.4. Formato de intercambio FBX

El formato FBX (Filmbox) es utilizado para el intercambio de recursos tridimensionales entre aplicaciones de producción 3D y motores gráficos. Este formato permite transferir información relacionada con la geometría de los modelos y, dependiendo de las características del recurso, puede incluir jerarquías, estructuras de huesos y datos de animación [5], [1].

Dentro de un pipeline de producción tridimensional, el uso de un formato de intercambio permite trasladar los recursos desde las herramientas empleadas para su creación hacia el entorno donde serán integrados. Sin embargo, la transferencia de un modelo no implica necesariamente que todos los elementos visuales queden configurados de manera automática en la aplicación de destino.

Las texturas y materiales pueden requerir archivos independientes y una configuración posterior dentro del motor gráfico, debido a que su representación depende de la organización de los mapas de textura, los materiales y las características del sistema de renderizado utilizado. Por esta razón, el proceso de intercambio debe considerar tanto la transferencia del modelo tridimensional como la preparación de los recursos complementarios necesarios para su correcta utilización [5], [1].

De esta manera, FBX constituye un formato de intercambio que facilita la transferencia de información tridimensional entre las diferentes herramientas que intervienen en un pipeline de producción, manteniendo separadas, cuando corresponde, las etapas de exportación del modelo y configuración de sus recursos visuales.

2.5.5. Optimización geométrica para aplicaciones en tiempo real

La optimización geométrica en recursos tridimensionales destinados a aplicaciones en tiempo real comprende procedimientos orientados a reducir y reorganizar la complejidad de la geometría sin eliminar las características visuales necesarias para la representación del modelo. Dentro de este proceso pueden utilizarse técnicas como la retopología, mediante la cual se obtiene una nueva malla con una distribución más controlada de polígonos y una estructura adecuada para las etapas posteriores de producción [20], [2], [1].

Uno de los flujos utilizados para este propósito corresponde al desarrollo High Poly–Low Poly. En este procedimiento, el modelo High Poly concentra una mayor cantidad de

información geométrica y detalles de superficie, mientras que posteriormente se genera una versión Low Poly mediante retopología con una estructura geométrica simplificada y preparada para su utilización dentro de aplicaciones interactivas.

La transferencia de determinados detalles visuales desde el modelo High Poly hacia la versión Low Poly puede realizarse mediante técnicas de baking. Este proceso permite generar mapas que contienen información de la superficie del modelo de mayor detalle, entre ellos Normal, Ambient Occlusion, Curvature, World Space Normal, Position y Thickness, los cuales pueden ser utilizados posteriormente durante el proceso de texturizado.

El Normal Map permite representar variaciones y detalles de superficie mediante información de normales sin necesidad de conservar toda la geometría presente en el modelo High Poly. De esta manera, la geometría y el detalle visual pueden ser gestionados de forma separada dentro del pipeline de producción.

No todos los recursos tridimensionales requieren necesariamente un flujo High Poly–Low Poly. Determinados objetos pueden desarrollarse directamente con una geometría de menor complejidad y complementar sus características visuales mediante materiales y mapas de textura, dependiendo de los requerimientos técnicos y artísticos del recurso.

En este contexto, la optimización geométrica mediante retopología y baking permite reducir y reorganizar la complejidad de la malla y transferir información visual hacia versiones de menor densidad geométrica. Sin embargo, estos procedimientos por sí solos no constituyen una demostración de mejora del rendimiento general de una aplicación, ya que factores como la tasa de fotogramas, el consumo de memoria o las llamadas de dibujo requieren mediciones específicas dentro del motor gráfico [20], [2], [1].

2.6. Herramientas de desarrollo

Para el desarrollo de los assets tridimensionales del videojuego InkFall se utilizaron diferentes herramientas de software de acuerdo con las necesidades de cada etapa del pipeline de producción. Cada una cumplió una función específica dentro de los procesos de planificación, modelado, texturizado, rigging, animación, exportación e integración de los recursos visuales.

Con el propósito de documentar el entorno tecnológico empleado durante el desarrollo de los assets tridimensionales de InkFall, se presenta la **Tabla 2**, en la cual se identifican las principales herramientas utilizadas, su versión de trabajo, la función desempeñada dentro del pipeline y los formatos empleados para el intercambio de recursos.

Tabla 2

Entorno técnico utilizado durante la producción de assets de InkFall

Herramienta	Versión utilizada	Función dentro del proyecto	Formatos principales
Tripo AI	Plataforma web	Generación de modelos tridimensionales base para determinados personajes, utilizados posteriormente como punto de partida para el proceso de producción en Blender.	.obj
Blender	5.0	Modelado 3D, escultura digital, retopología, mapeado UV, rigging, animación y preparación de modelos para exportación.	.blend / .fbx
Substance 3D Painter	12.1.2	Baking, creación de materiales PBR y texturizado de los assets.	.png
Unity	6000.3.10f1	Integración de modelos, configuración de materiales y texturas, incorporación de animaciones y comprobación de los assets dentro del videojuego.	.fbx / .png
Adobe Photoshop	27.9.1	Elaboración de representaciones preliminares y blocking de determinados escenarios.	.psd / archivos de imagen
Mixamo	Plataforma web	Auto-rigging y obtención de animaciones base para personajes humanoides.	.fbx

Nota. Las herramientas fueron empleadas de acuerdo con las necesidades específicas de cada etapa del pipeline. Los modelos tridimensionales fueron transferidos principalmente mediante archivos FBX, mientras que las texturas desarrolladas en Substance 3D Painter fueron exportadas en formato PNG e importadas posteriormente en Unity para la configuración de los materiales.

2.6.1. Blender

Blender fue la herramienta principal utilizada para la producción tridimensional. Mediante este software se realizaron procesos de modelado, escultura digital, retopología, mapeado UV, preparación de estructuras de rigging, animación y exportación de los assets. También fue utilizado para la construcción de escenarios, personajes, enemigos, armas, trampas y otros objetos necesarios para el videojuego. En el caso de recursos con estructuras no humanoides, como el enemigo de seis patas, Blender permitió desarrollar manualmente el sistema de huesos y las animaciones correspondientes. El propio documento evidencia el uso de Blender tanto para el rigging manual como para las animaciones del enemigo no humanoide [5], [4].

2.6.2. Substance 3D Painter

Substance 3D Painter fue utilizado para los procesos de baking, creación de materiales y texturizado de los assets. En los modelos desarrollados mediante un flujo High Poly–Low Poly se empleó esta herramienta para transferir información del modelo de alta resolución hacia el modelo de menor complejidad mediante la generación de mapas auxiliares. Posteriormente se desarrollaron los materiales y texturas correspondientes a cada recurso.

Una vez finalizado el proceso de texturizado, los mapas generados en Substance 3D Painter fueron exportados en formato PNG para su posterior incorporación al proyecto de Unity. De esta manera, las texturas se gestionaron de forma independiente a los archivos FBX utilizados para los modelos tridimensionales.

2.6.3. Mixamo

Mixamo se empleó como herramienta complementaria para el rigging y la animación de personajes humanoides. Los modelos compatibles fueron exportados para utilizar el sistema de auto-rigging de la plataforma y, posteriormente, el esqueleto generado fue revisado en Blender para realizar los ajustes necesarios. Mixamo también permitió seleccionar determinadas animaciones base requeridas por los personajes, como caminar, correr y permanecer en reposo.

2.6.4. Adobe Photoshop

Adobe Photoshop se utilizó como herramienta de apoyo durante las etapas iniciales de planificación de determinados escenarios. Su función dentro del proyecto estuvo relacionada con la elaboración de representaciones preliminares y propuestas de blocking que sirvieron como referencia para la posterior construcción tridimensional de los niveles.

2.6.5. Unity

Unity fue utilizado como motor gráfico para la integración y comprobación de los assets desarrollados. Dentro del motor se importaron los modelos tridimensionales, las texturas y las animaciones, se configuraron los materiales correspondientes y se revisaron aspectos como escala, orientación, ubicación y reproducción de las animaciones según el tipo de recurso. El proceso documentado muestra que las texturas generadas externamente fueron importadas y configuradas posteriormente dentro de Unity [6], [1].

Para la transferencia de la geometría, jerarquías, estructuras de huesos y animaciones desde Blender se utilizó principalmente el formato FBX, mientras que las texturas desarrolladas en Substance 3D Painter fueron exportadas mediante archivos PNG. Esta separación permitió organizar los distintos tipos de recursos según su función dentro del pipeline y realizar posteriormente su configuración dentro del motor gráfico [1], [5].

En conjunto, Blender, Substance 3D Painter, Mixamo, Adobe Photoshop y Unity conformaron el entorno de herramientas empleado durante la producción de los assets tridimensionales de InkFall. Su utilización respondió a las necesidades específicas de cada etapa del proceso y permitió establecer un flujo de trabajo desde la planificación inicial hasta la incorporación de los recursos dentro del videojuego.

2.7. Marco conceptual

2.7.1. Asset 3D

Un asset 3D es un recurso digital tridimensional utilizado dentro de un videojuego o aplicación interactiva. Puede representar personajes, escenarios, objetos, armas, mobiliario u otros elementos visuales que conforman el entorno del juego. Estos recursos son diseñados para ser integrados en motores gráficos y deben cumplir requisitos de calidad visual y optimización para garantizar un adecuado rendimiento [4], [1].

La producción de un asset puede involucrar diferentes etapas según sus características y función dentro del proyecto, entre ellas modelado tridimensional, escultura digital, retopología, mapeado UV, texturizado, rigging y animación. No todos los recursos requieren necesariamente la totalidad de estas etapas, debido a que el procedimiento depende del tipo de elemento desarrollado y de los requerimientos establecidos para su utilización.

Dentro de la presente Propuesta Tecnológica, el término asset 3D se utiliza para identificar los recursos tridimensionales desarrollados, preparados o integrados como parte del videojuego InkFall, incluyendo personajes, enemigos, escenarios y objetos requeridos para la construcción visual de los niveles.

2.7.2. Modelado tridimensional (3D Modeling)

El modelado tridimensional es el proceso de crear representaciones digitales de objetos mediante la construcción de geometrías en un espacio de tres dimensiones. Durante

este proceso se definen la forma, el volumen y la estructura del modelo utilizando vértices, aristas y polígonos, constituyendo la base para las etapas posteriores de texturizado y animación [4], [22].

El proceso de modelado puede variar de acuerdo con el tipo de asset y con el nivel de detalle requerido. En la producción de videojuegos pueden emplearse diferentes procedimientos para desarrollar personajes, escenarios, objetos y elementos interactivos, considerando posteriormente etapas complementarias como retopología, mapeado UV, texturizado, rigging o animación según las características del recurso.

Dentro del desarrollo de InkFall, el modelado tridimensional constituyó una de las etapas principales para la creación de personajes, enemigos, escenarios, armas, trampas y otros elementos visuales utilizados en los diferentes niveles del videojuego.

2.7.3. Escultura digital

La escultura digital consiste en la creación o modificación de modelos tridimensionales utilizando herramientas que simulan técnicas tradicionales de escultura. Este proceso permite añadir detalles orgánicos, mejorar la anatomía de personajes y generar superficies complejas antes de realizar la optimización del modelo para su utilización en videojuegos [13], [23].

En la producción de personajes para videojuegos, esta técnica puede utilizarse para desarrollar una versión de alta resolución o High Poly, en la cual se prioriza la definición visual del recurso antes de reducir y reorganizar su geometría para las etapas posteriores del pipeline.

Dentro del desarrollo de InkFall, la escultura digital se empleó principalmente durante la creación de personajes. A partir de los modelos base se realizaron modificaciones en la anatomía, proporciones, vestimenta y detalles de la superficie, obteniendo modelos High

Poly que posteriormente fueron utilizados como referencia para el proceso de retopología y para la generación de información mediante baking.

2.7.4. Retopología

La retopología es el proceso mediante el cual se crea o reorganiza la geometría de un modelo tridimensional con el propósito de obtener una malla con menor complejidad y una distribución de polígonos más adecuada para las etapas posteriores de producción.

Este procedimiento es especialmente utilizado cuando se parte de un modelo High Poly generado mediante escultura digital. A partir de este modelo se construye una nueva versión Low Poly que conserva la forma general del recurso mediante una cantidad reducida de geometría y una topología organizada de acuerdo con las necesidades del asset.

En personajes destinados a procesos de rigging y animación, la distribución de la topología también debe considerar las zonas que experimentarán deformaciones, como articulaciones y áreas de movimiento, con el propósito de facilitar el comportamiento de la malla durante la reproducción de las animaciones [20], [5].

Dentro del desarrollo de InkFall, la retopología se aplicó después de la etapa de escultura de los personajes para transformar los modelos High Poly en versiones Low Poly. Posteriormente, estas versiones fueron utilizadas para realizar el mapeado UV, el baking, el texturizado, el rigging y las demás etapas necesarias antes de su integración dentro del videojuego.

2.7.5. Mapeado UV (UV Mapping)

El mapeado UV es el proceso mediante el cual la superficie de un modelo tridimensional se despliega sobre un espacio bidimensional, generando coordenadas que

permiten relacionar cada zona de la geometría con una determinada región de una textura [5], [4].

Durante este proceso, la superficie del modelo se divide en diferentes secciones o islas UV, cuya distribución debe organizarse de manera que permita aprovechar adecuadamente el espacio disponible y reducir problemas como estiramientos, superposiciones o deformaciones visibles durante la aplicación de las texturas.

Dentro de un pipeline de producción tridimensional, el mapeado UV se realiza después de que la geometría principal del recurso ha sido definida y antes de procesos como el baking y el texturizado. Esta preparación permite que los mapas generados y las texturas desarrolladas puedan aplicarse correctamente sobre la superficie del modelo.

En el desarrollo de InkFall, el mapeado UV fue aplicado tanto en personajes como en escenarios y objetos tridimensionales, utilizando diferentes procedimientos según las características de cada recurso. Una vez preparadas las coordenadas UV, los modelos continuaron hacia las etapas de baking y texturizado en Substance 3D Painter cuando correspondía.

2.7.6. Texturizado

El texturizado es el proceso mediante el cual se aplican características visuales a la superficie de un modelo tridimensional con el propósito de representar colores, materiales, desgaste, rugosidad, detalles superficiales y otras propiedades necesarias para definir su apariencia [11], [2].

Este proceso se realiza sobre modelos que previamente cuentan con una distribución de coordenadas UV, lo que permite relacionar la superficie tridimensional con los diferentes mapas de textura utilizados durante la producción. Dependiendo del flujo de trabajo

empleado, pueden utilizarse mapas como Base Color, Normal, Roughness, Metallic y Ambient Occlusion, entre otros.

En la producción de assets para InkFall, el texturizado se realizó principalmente mediante Substance 3D Painter, donde se utilizaron materiales y mapas generados durante el proceso de baking para desarrollar la apariencia visual de personajes, escenarios y objetos. En esta etapa también se realizaron ajustes específicos según el tipo de superficie representada, como tela, cuero, metal, madera o piedra.

Una vez finalizado el texturizado, los mapas correspondientes fueron exportados en formato PNG e importados posteriormente a Unity, donde se asignaron a los materiales utilizados dentro del videojuego. De esta manera, el proceso de texturizado se mantuvo separado de la exportación de la geometría realizada mediante archivos FBX.

2.7.7. Physically Based Rendering (PBR)

El Physically Based Rendering (PBR) es un enfoque de representación de materiales que busca describir el comportamiento visual de las superficies a partir de propiedades relacionadas con la interacción de la luz. Para ello utiliza diferentes mapas de textura que almacenan información específica sobre el material.

Entre los mapas utilizados dentro de este tipo de flujo se encuentran Base Color, que define el color principal de la superficie; Roughness, que controla el grado de rugosidad; Metallic, que determina el comportamiento metálico del material; y Normal, que permite representar detalles superficiales sin modificar directamente la geometría del modelo. También pueden emplearse mapas complementarios como Ambient Occlusion [11], [2].

La utilización de un flujo PBR permite organizar de manera estructurada las propiedades visuales de un material; sin embargo, su apariencia final puede variar entre diferentes programas o motores gráficos. Factores como el shader empleado, la iluminación,

el render pipeline, la interpretación de los canales y la configuración de las texturas influyen en el resultado visual final.

Dentro del desarrollo de InkFall, este enfoque fue utilizado durante el proceso de texturizado en Substance 3D Painter. Posteriormente, los mapas generados fueron exportados e importados en Unity, donde se configuraron los materiales correspondientes según las necesidades visuales de cada asset.

2.7.8. Rigging

El rigging es el proceso mediante el cual se incorpora una estructura ósea o esqueleto digital a un modelo tridimensional con el propósito de permitir su posterior animación. Esta estructura se encuentra compuesta por huesos organizados jerárquicamente que controlan las diferentes partes del modelo [24], [13].

Durante este proceso también se establece la relación entre la geometría y el sistema óseo mediante la asignación de pesos de influencia, lo que determina la forma en que la malla se deforma cuando los huesos son desplazados o rotados durante una animación.

El procedimiento de rigging puede variar según la anatomía y las características del recurso. En personajes humanoides pueden utilizarse sistemas automáticos de generación de esqueletos, mientras que los modelos con estructuras no convencionales pueden requerir la creación y configuración manual de los huesos.

Dentro del desarrollo de InkFall, los personajes humanoides fueron preparados mediante el sistema de auto-rigging de Mixamo. Posteriormente, los esqueletos generados fueron revisados en Blender y, cuando fue necesario, se realizaron ajustes en la posición de los huesos y en los pesos de influencia. En el caso del enemigo no humanoide de seis patas, el sistema de rigging fue desarrollado manualmente en Blender debido a que su estructura corporal no era compatible con el sistema automático de Mixamo.

2.7.9. Baking

El baking es una técnica utilizada para transferir información de un modelo de alta resolución a otro optimizado mediante la generación de mapas de textura, como Normal Map o Ambient Occlusion. Este procedimiento permite conservar detalles visuales complejos sin incrementar significativamente la cantidad de polígonos del modelo [11], [2].

Dentro del desarrollo de InkFall, el proceso de baking se realizó en Substance 3D Painter, utilizando el modelo Low Poly como geometría principal y el modelo High Poly como fuente de información. A partir de este procedimiento se generaron diferentes mapas auxiliares utilizados durante el texturizado.

Entre los mapas generados se encuentran Normal Map, Ambient Occlusion, Curvature, World Space Normal, Position, Thickness e ID. Cada uno almacena información específica relacionada con la superficie, orientación, profundidad, posición o separación de materiales del modelo, y sirve como apoyo para la aplicación de materiales y efectos durante el proceso de texturizado.

El Normal Map permitió representar detalles de superficie provenientes del modelo High Poly sobre la versión Low Poly, mientras que los demás mapas fueron utilizados principalmente como información auxiliar dentro de Substance 3D Painter para facilitar la aplicación de materiales, máscaras, variaciones de desgaste y otros efectos visuales.

2.7.10. Motor gráfico (Game Engine)

Un motor gráfico es una plataforma de software que proporciona las herramientas necesarias para desarrollar videojuegos, incluyendo sistemas de renderizado, física, animación, audio y programación. En esta propuesta tecnológica se utilizó Unity como motor gráfico para integrar y visualizar los assets tridimensionales desarrollados durante el proyecto [1], [6].

2.7.11. Blender

Blender es un software libre y de código abierto especializado en la creación de contenido tridimensional. Ofrece herramientas para modelado, escultura digital, animación, retopología, mapeado UV, renderizado y exportación de modelos, siendo ampliamente utilizado en la producción de videojuegos y contenidos digitales [5], [4].

Dentro de un pipeline de producción para videojuegos, Blender puede utilizarse en diferentes etapas según las características del asset, desde la construcción inicial de la geometría hasta la preparación de sistemas de huesos y animaciones.

En el desarrollo de InkFall, Blender fue empleado como una de las herramientas principales para la creación de personajes, enemigos, escenarios, armas, trampas y otros recursos tridimensionales. También se utilizó para realizar procesos de escultura High Poly, retopología Low Poly, mapeado UV y preparación de modelos antes de su exportación.

En el caso del enemigo no humanoide de seis patas, Blender fue utilizado además para desarrollar manualmente el sistema de rigging y las animaciones, debido a que la estructura del personaje no era compatible con el sistema automático de Mixamo.

2.7.12. Unity

Unity es un motor de desarrollo de videojuegos que permite integrar y gestionar diferentes tipos de recursos dentro de un entorno interactivo. Entre sus funciones se encuentran la organización de escenas, incorporación de modelos tridimensionales, configuración de materiales y texturas, reproducción de animaciones, gestión de componentes y ejecución de pruebas dentro del proyecto [6], [1].

En un pipeline de producción tridimensional, Unity funciona como el entorno donde los assets creados mediante herramientas externas son incorporados y configurados para su utilización dentro del videojuego. Este proceso puede incluir la importación de geometría,

asignación de materiales, configuración de texturas, preparación de animaciones y revisión de aspectos técnicos relacionados con la escala, orientación y ubicación de los recursos.

Dentro del desarrollo de InkFall, Unity fue utilizado para integrar los modelos tridimensionales exportados principalmente en formato FBX y las texturas generadas en Substance 3D Painter en formato PNG. Posteriormente, se configuraron los materiales correspondientes y se revisó la correcta visualización de los recursos dentro de las escenas del videojuego.

Asimismo, Unity permitió comprobar el funcionamiento de los assets animados, incluyendo la reproducción de animaciones y la integración de los sistemas de rigging cuando correspondía. Estas verificaciones formaron parte del proceso de incorporación de los recursos tridimensionales desarrollados al entorno funcional de InkFall.

2.7.13. Pipeline de producción de assets 3D

Un pipeline de producción de assets 3D corresponde a una secuencia organizada de etapas y procedimientos utilizados para desarrollar, preparar y transferir recursos tridimensionales desde su planificación inicial hasta su incorporación dentro de un motor gráfico. Dependiendo de las características del recurso, este flujo puede incluir procesos como modelado tridimensional, escultura digital, retopología, mapeado UV, baking, texturizado, rigging, animación, exportación e integración [1].

La estructura de un pipeline permite establecer relaciones entre las distintas fases del proceso de producción, debido a que los resultados obtenidos en una etapa pueden constituir la base necesaria para continuar con las siguientes. Sin embargo, el flujo puede adaptarse según el tipo de asset, ya que no todos los recursos requieren necesariamente la totalidad de los procedimientos.

Dentro de la presente propuesta tecnológica, el término pipeline de producción se utiliza para hacer referencia al flujo de trabajo aplicado para organizar el desarrollo, preparación e integración de los assets tridimensionales destinados al videojuego InkFall.

2.8. Relación entre la teoría y la propuesta tecnológica

Los fundamentos teóricos, metodológicos y tecnológicos abordados en el presente capítulo proporcionan la base conceptual para el desarrollo de los assets tridimensionales del videojuego InkFall. Los conceptos relacionados con modelado 3D, escultura digital, retopología, mapeado UV, baking, texturizado, rigging y animación permiten comprender las diferentes etapas que intervienen en la preparación de recursos tridimensionales destinados a un entorno interactivo [2], [4].

La revisión de los procesos de producción de assets permitió establecer un pipeline estructurado como referencia para organizar las actividades desarrolladas durante el proyecto. Este flujo contempla diferentes etapas que pueden adaptarse de acuerdo con las características de cada recurso, considerando que personajes, escenarios y objetos no necesariamente requieren los mismos procedimientos de producción [3], [21].

Los fundamentos relacionados con la escultura High Poly, la retopología y la generación de modelos Low Poly sustentan el proceso de optimización geométrica aplicado a determinados assets. De manera complementaria, el mapeado UV y el baking permiten preparar los modelos para el desarrollo de materiales y texturas, conservando visualmente determinados detalles del modelo de alta resolución mediante mapas de textura [2], [4], [20].

Los principios del Physically Based Rendering (PBR) proporcionan la base para la creación y organización de los materiales utilizados en los assets. Su aplicación mediante Substance 3D Painter permite trabajar distintas propiedades visuales de las superficies y generar mapas que posteriormente pueden ser configurados dentro del motor gráfico. Sin

embargo, la apariencia final de estos materiales depende de su configuración en Unity, por lo que la integración requiere una revisión posterior de las texturas y materiales utilizados [6], [11].

De igual manera, los conceptos relacionados con rigging y animación fundamentan la preparación de los personajes y elementos que requieren movimiento. La selección de los procedimientos aplicados depende de las características estructurales de cada recurso, permitiendo utilizar diferentes alternativas de acuerdo con las necesidades del proyecto [4], [13].

En el ámbito tecnológico, herramientas como Blender, Substance 3D Painter, Mixamo, Adobe Photoshop y Unity cumplen funciones específicas dentro del proceso de producción. Asimismo, el uso del formato FBX para la transferencia de modelos tridimensionales y del formato PNG para las texturas permite diferenciar los distintos tipos de información utilizados durante la integración de los recursos [5], [6].

En conjunto, los contenidos desarrollados en el marco teórico permiten relacionar los conceptos estudiados con las decisiones adoptadas durante la producción de los assets de InkFall. Estos fundamentos sirven como base para comprender el proceso técnico que se documenta posteriormente en el Manual Técnico de Implementación, donde se presentan de manera detallada los procedimientos aplicados durante el desarrollo e integración de los recursos tridimensionales.

Capítulo III

3. Manual del usuario

3.1. Introducción

El presente Manual de Usuario tiene como finalidad proporcionar la información necesaria para la correcta instalación, ejecución y utilización del prototipo del videojuego Inkfall. Este documento está dirigido a cualquier usuario que desee interactuar con el sistema, ofreciendo una guía clara sobre los requisitos del software, los controles, las mecánicas principales y el funcionamiento general del juego.

Inkfall es un videojuego narrativo desarrollado en el motor Unity como parte de un proyecto académico. El prototipo está compuesto por dos niveles que presentan diferentes mecánicas de juego y un sistema de dificultad dinámica que modifica determinadas condiciones de la partida en función del desempeño del jugador.

Este manual describe paso a paso el proceso para ejecutar el videojuego, conocer la interfaz, utilizar los controles y comprender las principales mecánicas implementadas.

3.2. Requisitos del sistema

Tabla 3

Requisitos mínimos del sistema

Componente	Requisitos
Sistema Operativo	Windows 10 de 64 bits
Procesador	Intel Core i5 de 8va generación o AMD Ryzen 5 equivalente
Memoria RAM	8 GB
Tarjeta gráfica	NVIDIA GTX 1650 o AMD Radeon RX 570, con 4 GB de VRAM
Almacenamiento	8 GB disponibles

Nota. Elaboración propia

3.3. Instalación

Para ejecutar el videojuego Inkfall es necesario contar con un equipo que cumpla con los requisitos mínimos del sistema y seguir el procedimiento de instalación descrito a continuación.

3.3.1. Descarga del juego

Obtenga el archivo comprimido que contiene el videojuego desde el medio proporcionado por los desarrolladores (memoria USB, enlace de descarga o repositorio institucional).

3.3.2. Descompresión de los archivos

Ubique el archivo descargado y descomprímalo en la carpeta de su preferencia utilizando el explorador de archivos de Windows o cualquier software de descompresión compatible.

3.3.3. Ejecución del juego

Abra la carpeta del proyecto y ejecute el archivo denominado Inkfall.exe para iniciar el videojuego.

3.3.4. Inicio del videojuego

Una vez ejecutado el archivo, se mostrará la pantalla principal del videojuego, desde la cual el usuario podrá acceder al menú principal y comenzar la experiencia de juego.

3.4. Interfaz del juego

La interfaz de usuario de Inkfall fue diseñada para ofrecer una navegación clara e intuitiva desde el inicio del videojuego. Su estructura permite acceder de forma organizada a las diferentes funciones disponibles antes de comenzar la partida, manteniendo una

distribución uniforme de los elementos visuales y una identidad gráfica coherente con la propuesta artística del proyecto.

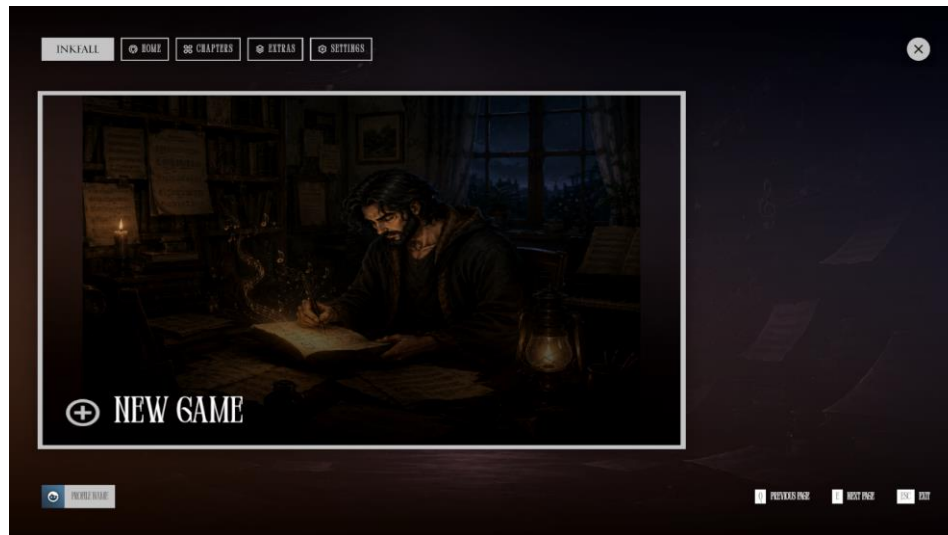
Al ejecutar el videojuego, el usuario visualizará una pantalla de bienvenida con el logotipo de Inkfall y el mensaje "Press Any Key to Start". Para acceder al menú principal, únicamente deberá presionar cualquier tecla del teclado.

3.4.1. Menú principal

El menú principal constituye el centro de navegación del videojuego y reúne las principales opciones disponibles para el usuario. La interfaz está organizada mediante pestañas ubicadas en la parte superior, permitiendo cambiar entre las diferentes secciones sin abandonar el menú principal.

Ilustración 2

Menú principal.



Nota. Menú de inicio del juego, Elaboración propia

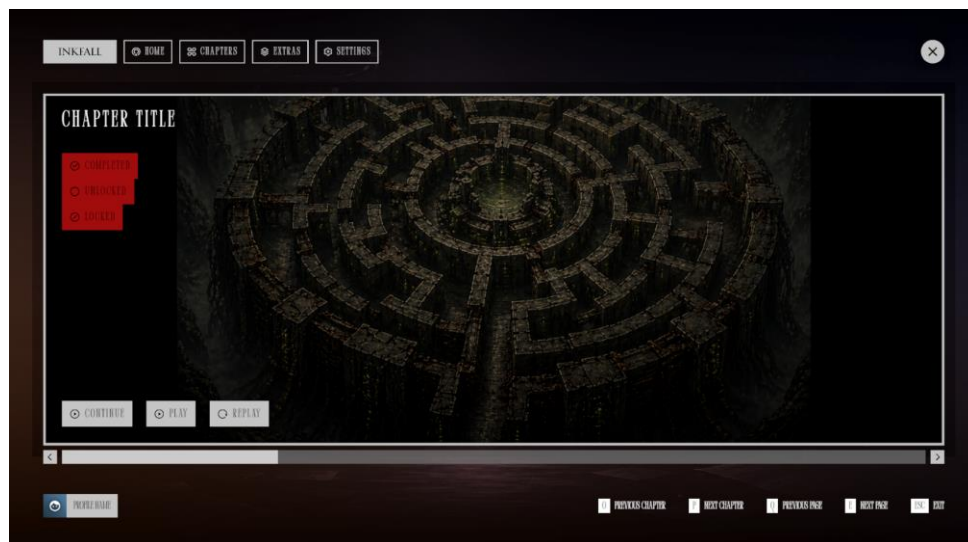
La pestaña Home presenta la opción New Game, desde donde el usuario puede iniciar una nueva partida y acceder al contenido del videojuego. Además, en la parte inferior se

muestran los controles disponibles para desplazarse entre las páginas del menú o salir de la aplicación.

La pestaña Chapters permite visualizar los capítulos disponibles dentro del prototipo. Cada capítulo incluye una imagen representativa, una breve descripción y las opciones para iniciar o repetir el nivel correspondiente.

Ilustración 3

Pestaña de Chapters.



Nota. Elaboración propia.

La pestaña Extras contiene información complementaria del videojuego, como los créditos.

Ilustración 4

Pestaña de extras.

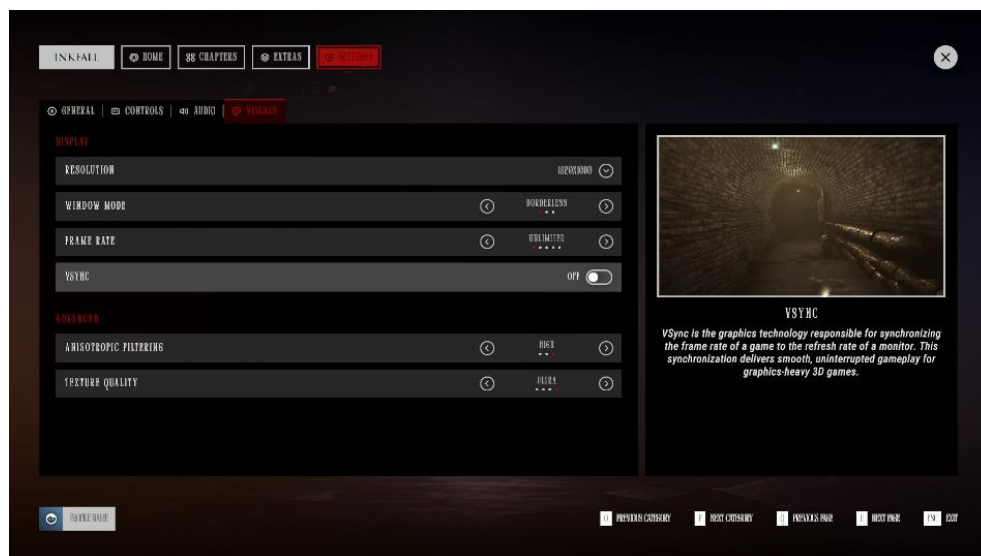


Nota. Elaboración propia.

Finalmente, la pestaña Settings permite modificar diferentes parámetros del videojuego. Dentro de esta sección el usuario puede acceder a opciones generales, configuración de controles, audio, aspectos visuales y funciones de accesibilidad, adaptando la experiencia de juego de acuerdo con sus preferencias.

Ilustración 5

Pestaña de Settings.



Nota. Elaboración propia.

3.4.2. Pantalla del juego

Durante la partida se mostrará la interfaz principal (HUD), la cual proporciona al jugador la información necesaria para el desarrollo del juego. Su diseño mantiene la identidad visual de InkFall, utilizando elementos gráficos inspirados en el logotipo del videojuego.

El indicador de vida está representado por el círculo presente en el logotipo, el cual refleja el estado de salud del personaje durante la partida. En el centro se encuentra una gota que indica el nivel de dificultad seleccionado, permitiendo al jugador identificar esta información de forma rápida sin afectar la experiencia visual.

En el Capítulo 1, la interfaz incorpora información adicional relacionada con el sistema de combate, mostrando el arma equipada, el inventario de armas y la cantidad de munición disponible. Estos elementos permiten al jugador administrar sus recursos durante los enfrentamientos.

Por otra parte, en el Capítulo 2, la interfaz se adapta a la mecánica propia de ese escenario. Debido a que la jugabilidad está basada en el uso de una vela como elemento principal, los indicadores relacionados con las armas son reemplazados por la información necesaria para esta mecánica, manteniendo una interfaz clara y acorde con la experiencia del jugador

3.5. Controles

Tabla 4

Controles

Acción	Tecla
Mover	WASD
Cámara	Mouse
Disparar	Click izquierdo
Apuntar	Click derecho
Saltar	Espacio
Interactuar	E
Recargar	R
Habilidad especial	F
Cambiar arma	Rueda del mouse
Abrir puerta	Click izquierdo
Recargar encendedor	R

Nota. Elaboración propia.

3.6. Como jugar

Al iniciar Inkfall, el jugador tomará el control del protagonista en 2 escenarios distintos. El objetivo es explorar los escenarios, interactuar con los elementos del entorno y superar los desafíos planteados en cada nivel. A medida que el jugador progresa, encontrará mecánicas diferentes que requieren combinar exploración, observación y habilidad para completar cada sección del prototipo.

3.6.1. Inicio

Al ejecutar el juego se mostrará el menú principal, desde donde el usuario podrá iniciar una nueva partida o seleccionar uno de los 2 niveles. Una vez seleccionada la opción correspondiente, el juego cargará el nivel y el jugador aparecerá en el punto inicial del escenario.

Durante la partida, el jugador podrá desplazarse libremente utilizando los controles establecidos e interactuar con los diferentes elementos presentes en el entorno para avanzar en la experiencia.

3.6.2. Primer nivel

El primer nivel está orientado a las mecánicas de combate y exploración. El jugador deberá recorrer un laberinto mientras enfrenta enemigos controlados por inteligencia artificial.

Para avanzar es necesario:

- Explorar el escenario para encontrar la ruta correcta
- Utilizar el arma para eliminar a los enemigos que bloquean el camino
- Evitar recibir demasiado daño durante los enfrentamientos
- Continuar avanzando hasta alcanzar la salida del nivel

Si la vida del jugador llega a cero, la partida finalizará y el juego se reiniciará aplicando el sistema de dificultad dinámica.

3.6.3. Segundo nivel

El segundo nivel cambia completamente la dinámica del juego. En este escenario, la supervivencia depende del manejo adecuado de una vela que actúa como la principal fuente de luz del jugador, así el nivel transforma el juego a un nivel de terror.

Durante el recorrido, la vela puede apagarse debido a movimientos bruscos ya sea de la cámara o del personaje. Cuando esto ocurre, el jugador deberá encenderla lo antes posible para evitar permanecer en la oscuridad durante un tiempo prolongado.

Mientras explora el escenario, el jugador deberá recorrer las habitaciones, interactuar con los objetos necesarios y encontrar la ruta y la llave que le permita completar el nivel evitando ser alcanzado por la amenaza presente en el escenario.

3.6.4. Sistema de dificultad

Inkfall incorpora un sistema de dificultad dinámica que aumenta progresivamente el comportamiento del juego conforme el jugador acumula intentos fallidos.

Cada vez que el jugador muere, el sistema registra el intento y aplica cambios en determinados elementos del juego. Estos cambios pueden afectar el comportamiento de los enemigos, las condiciones del escenario o las mecánicas disponibles, generando una experiencia diferente en cada nuevo intento.

Este sistema forma parte de la propuesta principal del prototipo y busca demostrar una alternativa a los modelos tradicionales de dificultad basados únicamente en el incremento de parámetros numéricos o en la disminución de la dificultad por fracaso para hacer más fácil el nivel.

3.7. Créditos

Tabla 5

Integrantes del proyecto y su rol.

Integrante	Rol
Joan Dután	Programador principal
Alejandro Ortiz	Diseño y arte visual
Juan José Bacuilima	Modelado 3D y animación
Alexander Abad	Integración de sistemas

Elaboración propia.

Capítulo IV

4. Manual técnico de implementación

4.1. Producción de assets 3D

La producción de assets tridimensionales para el videojuego InkFall comprendió un conjunto de etapas destinadas a desarrollar, preparar e integrar personajes, enemigos, escenarios y objetos requeridos para los niveles incluidos dentro del prototipo. El procedimiento se organizó mediante un pipeline de producción artística adaptado a las características de cada tipo de recurso.

El proceso inició con la revisión del Game Design Document (GDD) y de los requerimientos establecidos por el equipo de desarrollo. A partir de esta información se identificaron los recursos necesarios para cada nivel y se recopiló referencias visuales relacionadas con su apariencia, función y relación con la dirección artística del videojuego.

Para los escenarios se realizaron representaciones preliminares mediante blocking como parte de la etapa de planificación. En el caso de Concret Loop, estas representaciones sirvieron como referencia para el posterior desarrollo tridimensional del laberinto en Blender. Para Crow, the Drowned Clown, el proceso se limitó a la elaboración de una propuesta preliminar de distribución, sin incluir el modelado tridimensional de dicho escenario dentro de la presente propuesta.

En el caso de determinados personajes, el flujo incluyó la creación de una versión High Poly mediante escultura digital, seguida por un proceso de retopología para obtener una versión Low Poly. Después se realizó el mapeado UV y el baking en Substance 3D Painter, permitiendo generar los mapas auxiliares utilizados durante el proceso de texturizado.

Los assets que requerían movimiento continuaron hacia las etapas de rigging y animación. Para los personajes humanoides se utilizó Mixamo como herramienta complementaria, mientras que los recursos con estructuras no humanoides requirieron procedimientos de rigging y animación desarrollados directamente en Blender.

Una vez preparados los recursos, los modelos tridimensionales fueron exportados principalmente en formato FBX, mientras que las texturas desarrolladas en Substance 3D Painter fueron exportadas por separado en formato PNG. Posteriormente, ambos tipos de archivos fueron importados en Unity, donde se configuraron los materiales y se realizaron las comprobaciones correspondientes a la geometría, escala, orientación y animaciones según las características de cada asset.

No todos los recursos desarrollados atravesaron la totalidad de las etapas descritas. El procedimiento fue adaptado según las necesidades de cada asset; por ejemplo, determinados props y elementos del escenario fueron modelados directamente con una geometría de menor complejidad, mientras que los personajes requirieron procesos adicionales de escultura, retopología, rigging y animación.

En conjunto, este flujo permitió organizar de forma estructurada la producción de los recursos tridimensionales desarrollados para InkFall y documentar las diferentes etapas necesarias para su posterior integración y evaluación dentro del motor gráfico Unity.

4.2. Pipeline de producción artística

4.2.1. Planificación e investigación de los assets

La primera etapa del proceso de implementación consistió en la planificación de los recursos tridimensionales que serían desarrollados para el videojuego InkFall. Antes de iniciar cualquier actividad de modelado fue necesario analizar las características generales

del proyecto, identificar los escenarios que conforman la experiencia de juego y determinar los personajes y objetos que debían producirse para cada nivel.

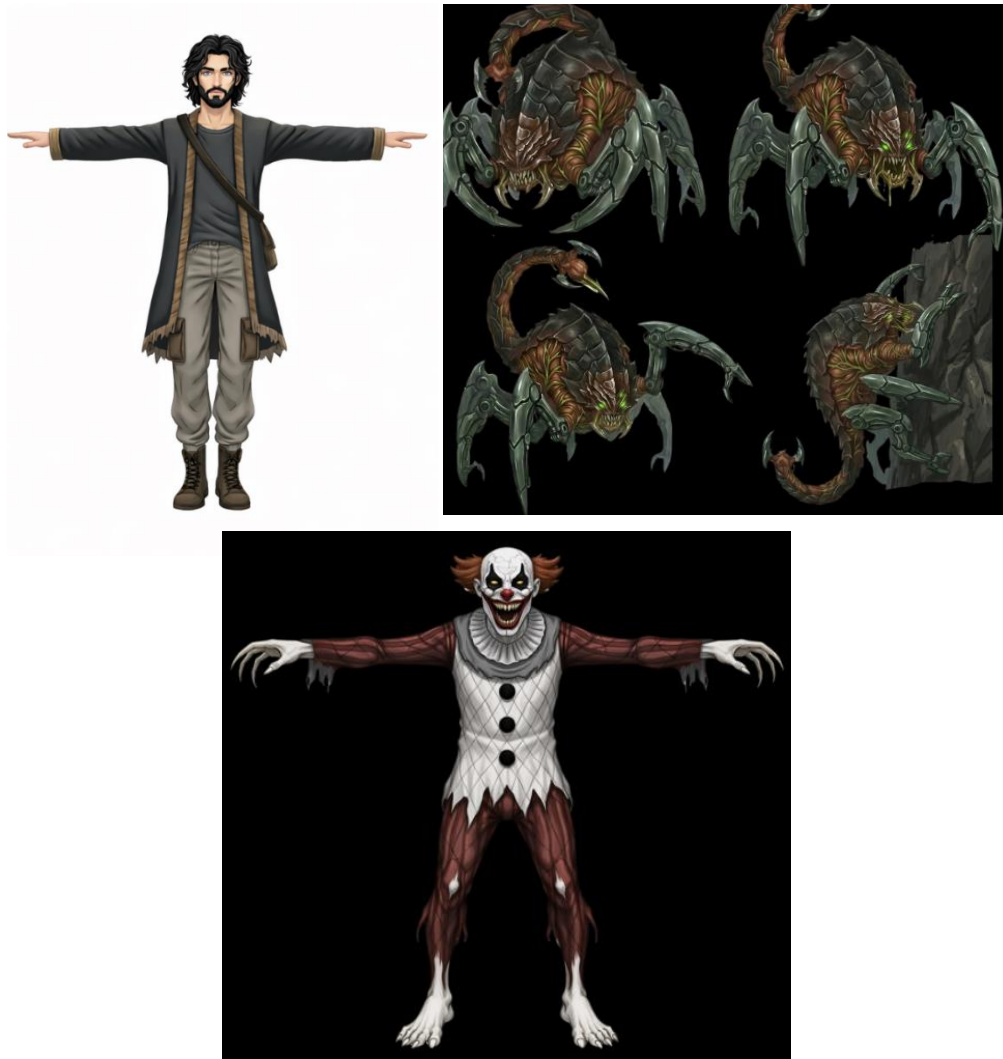
Como principal fuente de información se utilizó el Game Design Document (GDD), documento que reúne la información relacionada con la narrativa, las mecánicas de juego, la ambientación, el estilo artístico y las características de cada uno de los elementos que forman parte del videojuego. La revisión de este documento permitió conocer la apariencia visual de los personajes diseñados por el equipo de arte, así como las características de los escenarios y los objetos interactivos requeridos para el desarrollo del proyecto.

Durante esta etapa también se identificaron las necesidades específicas de cada nivel del videojuego. El primer nivel, denominado Concret Loop, corresponde a un escenario tipo shooter ambientado en un laberinto dinámico donde el jugador debe enfrentarse a diferentes criaturas mientras busca una salida. Por su parte, el segundo nivel, Crow, the Drowned Clown, presenta una experiencia de sigilo desarrollada dentro de una vivienda, en la que el jugador debe desplazarse utilizando una vela como fuente de iluminación mientras evita ser descubierto por el enemigo principal. Estas diferencias determinaron el tipo de assets que debían producirse y las características visuales de cada escenario.

Una vez definidos los requerimientos del proyecto, se realizó la recopilación y análisis de referencias visuales que sirvieron como apoyo para el desarrollo de personajes, escenarios y objetos interactivos. Este proceso permitió mantener una identidad visual coherente entre los diferentes recursos gráficos y facilitó la toma de decisiones durante las siguientes etapas del flujo de trabajo.

Ilustración 6

Referencias visuales de los personajes de InkFall



Nota. Referencias visuales utilizadas durante la planificación de los personajes de InkFall. Fuente: Game Design Document (GDD) del proyecto.

Ilustración 7

Referencias visuales para el diseño de armas



Nota. Referencias visuales utilizadas para definir las características generales de las armas del primer nivel. Fuente: Game Design Document (GDD) del proyecto.

Al finalizar esta etapa se obtuvo una planificación clara de los assets necesarios para el videojuego, identificando los personajes, escenarios y objetos que debían desarrollarse, así como las referencias visuales y los criterios técnicos que guiaron el resto del proceso de implementación.

4.2.2. Diseño preliminar de escenarios (Blocking)

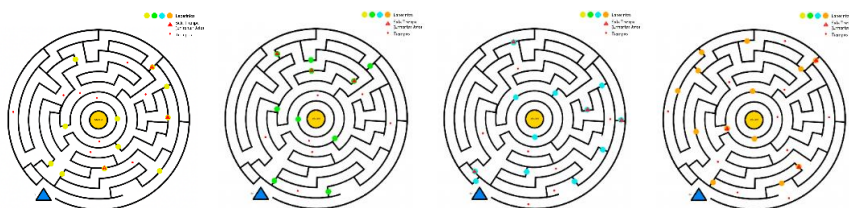
El desarrollo de los escenarios de InkFall inició con una etapa de diseño preliminar o blocking, cuyo propósito fue establecer de manera esquemática la distribución general de los espacios antes de comenzar su construcción tridimensional. Para esta etapa se utilizó Adobe Photoshop, permitiendo representar los recorridos principales, zonas de interacción y ubicación aproximada de diferentes elementos relacionados con las mecánicas de cada nivel.

Para el primer nivel, denominado Concret Loop, se realizaron cuatro configuraciones diferentes del laberinto. Durante esta planificación se definieron los recorridos principales, bifurcaciones, zonas destinadas a la aparición de trampas y

posibles rutas de desplazamiento del jugador. Estas variantes fueron planteadas considerando la mecánica del nivel, en la cual la estructura del laberinto puede modificarse durante la experiencia de juego.

Ilustración 8

Blocking del laberinto del nivel Concret Loop.



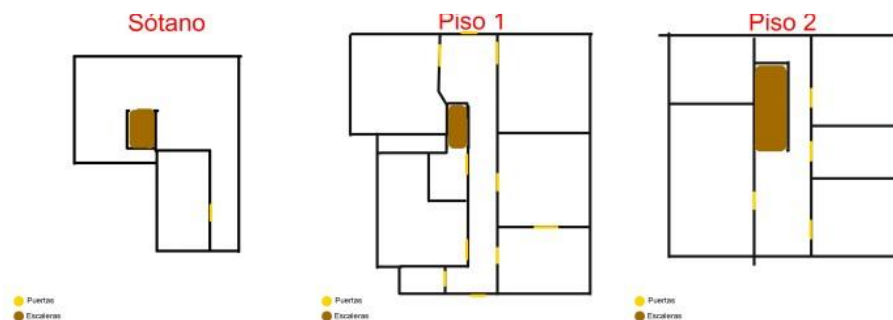
Nota. Captura del blocking de las cuatro configuraciones del laberinto elaborada en Adobe Photoshop. Elaboración propia.

Los esquemas desarrollados sirvieron como referencia visual para trasladar posteriormente la distribución planteada hacia el entorno tridimensional. Debido a que el blocking fue realizado como una representación preliminar, las proporciones y dimensiones definitivas debieron ajustarse posteriormente durante la construcción del escenario en Blender y su integración en Unity.

En el segundo nivel, denominado Crow, the Drowned Clown, el blocking se utilizó para establecer la distribución general de habitaciones, pasillos, escaleras y espacios relacionados con las mecánicas de exploración y sigilo. También se consideraron posibles ubicaciones para elementos interactivos y zonas necesarias para el recorrido del jugador dentro de la vivienda.

Ilustración 9

Blocking del escenario Crow, the Drowned Clown



Nota. Representación preliminar de la distribución espacial del segundo nivel elaborada durante la etapa de planificación. Elaboración propia.

La elaboración de estos diseños permitió contar con una referencia inicial para la construcción de los escenarios tridimensionales y facilitó la organización de los espacios antes de iniciar el modelado. Sin embargo, la distribución representada durante el blocking no fue considerada una estructura definitiva, debido a que durante las etapas posteriores fue necesario realizar ajustes relacionados con proporciones, escala y necesidades de navegación dentro de cada nivel.

Como resultado de esta etapa se obtuvieron cuatro configuraciones preliminares del laberinto de Concret Loop y una propuesta inicial de distribución para Crow, the Drowned Clown. En el primer caso, los esquemas fueron utilizados como referencia para el posterior desarrollo tridimensional del laberinto. En el segundo caso, el blocking se limitó a una propuesta preliminar de planificación y no continuó con una etapa de modelado tridimensional dentro de la presente propuesta.

4.2.3. Desarrollo de escenarios tridimensionales

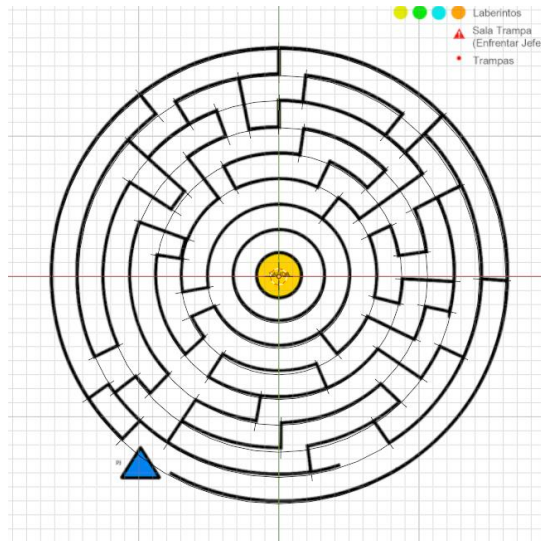
Una vez finalizada la etapa de planificación y blocking, se inició el desarrollo tridimensional del laberinto correspondiente al nivel Concret Loop.

Su construcción se realizó en Blender tomando como referencia los diseños preliminares elaborados durante la etapa de blocking. Para facilitar la construcción y posterior modificación de los recorridos se empleó un sistema basado en curvas.

Inicialmente se creó una curva que definía el recorrido general del laberinto. Los puntos de control fueron distribuidos siguiendo la estructura planteada previamente, de manera que la curva funcionara como eje para la generación posterior de las paredes.

Ilustración 10

Curva que define el recorrido inicial del laberinto.



Nota. Creación de las paredes del laberinto mediante curvas en Blender.

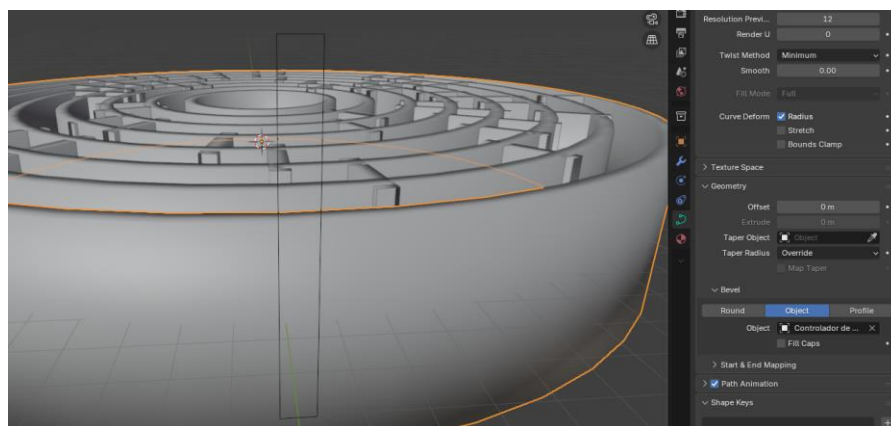
Para establecer la sección transversal de las paredes se creó un plano en Blender al que se eliminó la cara, conservando únicamente las aristas necesarias para definir un

perfil rectangular. Posteriormente, este elemento fue convertido en una curva mediante la opción Object → Convert → Curve.

El perfil obtenido fue asignado a la curva principal desde las propiedades de geometría, utilizando la configuración Properties → Geometry → Bevel → Object. De esta manera, la geometría de las paredes se generó siguiendo el recorrido establecido por la curva principal. Este procedimiento también permitió modificar posteriormente la forma del laberinto mediante la edición de sus puntos de control.

Ilustración 11

Configuración del perfil utilizado para generar las paredes del laberinto

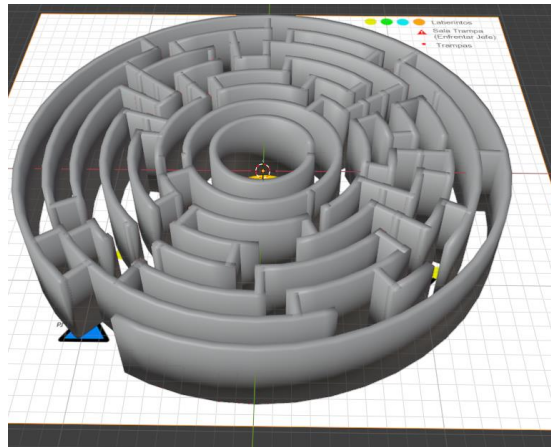


Nota. Configuración de la geometría de la curva mediante la opción Bevel Object en Blender.

Una vez establecida la distribución del escenario, la geometría generada mediante curvas fue convertida en una malla utilizando la opción Object → Convert → Mesh. Esta conversión permitió continuar con las etapas de preparación de la geometría, mapeado UV y texturizado.

Ilustración 12

Conversión de la curva a malla (Convert to Mesh).

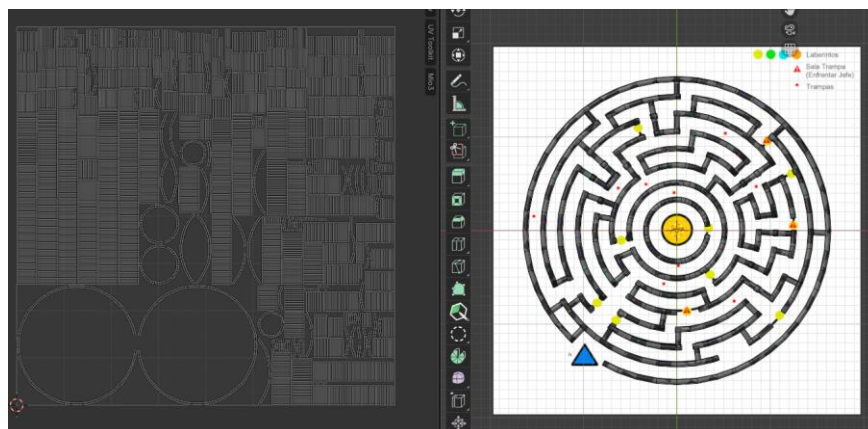


Nota. Conversión de la geometría generada mediante curvas a una malla editable en Blender. Elaboración propia.

Posteriormente se realizó el despliegue de coordenadas UV del laberinto mediante la herramienta Smart UV Project de Blender. Este procedimiento generó la distribución de las islas UV necesaria para continuar con la preparación de las superficies y la aplicación de las texturas.

Ilustración 13

Despliegue UV del laberinto mediante Smart UV Project



Nota. Distribución de coordenadas UV generada para preparar las superficies del laberinto para el texturizado.

Elaboración propia.

Como resultado de esta etapa se obtuvo el escenario tridimensional correspondiente al laberinto de Concret Loop, desarrollado mediante curvas y mallas en Blender y preparado para su posterior texturización e integración dentro del motor gráfico Unity.

4.2.4. Desarrollo de personajes tridimensionales

El desarrollo de los personajes tridimensionales de InkFall comenzó con la revisión de los diseños establecidos en el Game Design Document (GDD). Estos diseños fueron elaborados previamente por el equipo de arte y sirvieron como referencia para identificar características como la apariencia general, vestimenta, proporciones, colores predominantes y elementos distintivos de cada personaje.

A partir de estas referencias se utilizó Tripo AI como herramienta de apoyo para generar los modelos tridimensionales base de determinados personajes. El propósito de estos modelos fue proporcionar una geometría inicial sobre la cual continuar el proceso de producción, por lo que no fueron considerados recursos terminados ni modelos aptos para su integración directa en el videojuego.

Los modelos generados mediante Tripo AI presentaron diferentes inconvenientes en su geometría, entre ellos proporciones incorrectas, extremidades adicionales, irregularidades en la superficie y una distribución de la malla que no resultaba adecuada para continuar directamente con las etapas posteriores del pipeline.

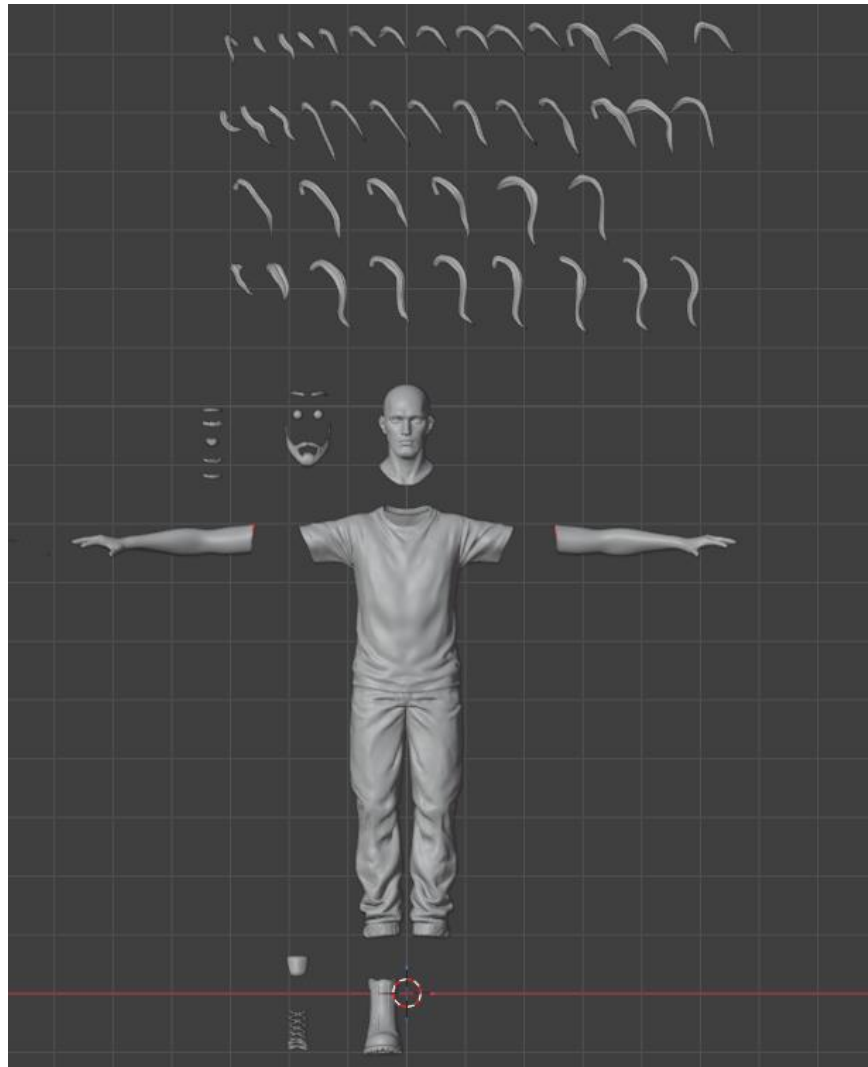
Por esta razón, los modelos generados fueron utilizados únicamente como punto de partida. Posteriormente se importaron en Blender, donde se inició su modificación para corregir las formas generales, proporciones y diferentes problemas presentes en la geometría original.

Esta fase tuvo como propósito modificar y refinar la geometría inicial para desarrollar con mayor precisión las características visuales de cada personaje.

Durante el proceso se realizaron ajustes sobre las formas generales del cuerpo, las proporciones anatómicas, el volumen de las extremidades y las características faciales. Asimismo, se trabajaron elementos correspondientes a la vestimenta, como pliegues y variaciones de superficie, de acuerdo con las referencias visuales establecidas previamente en el Game Design Document (GDD).

Ilustración 15

Proceso de escultura digital del personaje en Blender

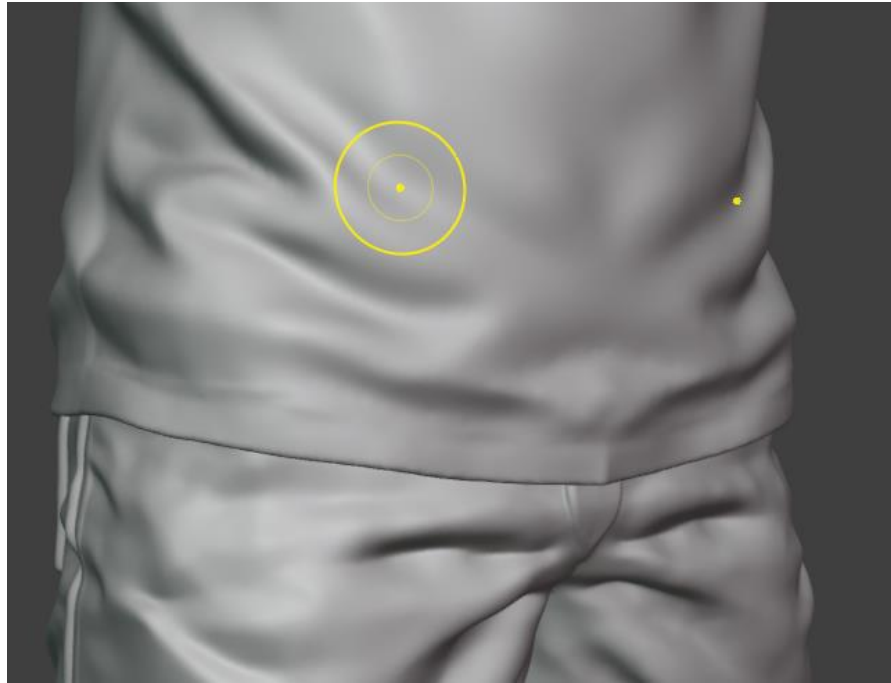


Nota. Modificación y refinamiento de la geometría del personaje mediante Sculpt Mode en Blender. Elaboración propia.

La escultura también permitió intervenir sobre diferentes problemas presentes en los modelos base generados mediante Tripo AI. Entre las principales correcciones realizadas se encontraron proporciones inadecuadas, irregularidades en determinadas zonas de la superficie y deformaciones que requerían ser reconstruidas antes de continuar con el pipeline.

Ilustración 16

Refinamiento de detalles mediante escultura digital



Nota. Detalle del proceso de escultura aplicado a los pliegues y formas superficiales del personaje. Elaboración propia.

Durante esta etapa se priorizó el desarrollo de la forma y del detalle geométrico, sin establecer todavía como criterio principal una cantidad reducida de polígonos. De esta manera se obtuvo una versión de alta resolución o High Poly, utilizada posteriormente como referencia para la construcción de una nueva malla mediante retopología y para determinados procesos de baking.

El trabajo de escultura se desarrolló de manera progresiva, comenzando por las formas generales del personaje y continuando posteriormente con detalles de menor escala. Durante este proceso se comparó el modelo con las referencias visuales del proyecto para mantener las características principales definidas para cada personaje.

Ilustración 17

Modelo High Poly finalizado.



Nota. Resultado de la etapa de escultura digital correspondiente a la versión High Poly del personaje. Elaboración propia.

Como resultado de esta etapa se obtuvieron las versiones High Poly de los personajes trabajados, con una geometría de mayor densidad y un nivel de detalle superior al de los modelos base iniciales. Estos modelos sirvieron como referencia para la siguiente etapa del pipeline, correspondiente a la retopología, en la que se desarrollaron las versiones Low Poly destinadas a continuar con la preparación técnica de los personajes.

4.2.6. Retopología

Una vez finalizada la escultura digital de los personajes, se realizó el proceso de retopología en Blender. Esta etapa consistió en construir una nueva geometría sobre la versión High Poly con el propósito de reducir su complejidad geométrica y obtener una malla con una distribución más organizada para continuar con las siguientes fases del pipeline.

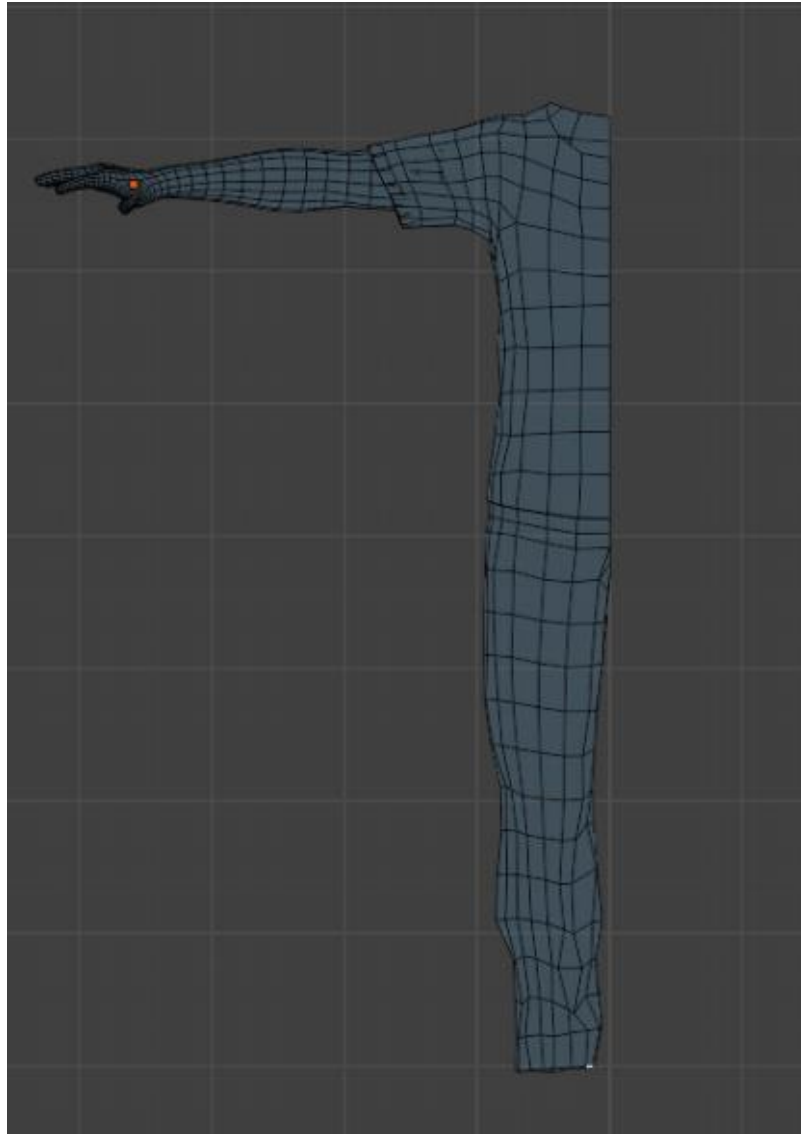
El modelo High Poly generado durante la escultura presentaba una elevada densidad de geometría debido al nivel de detalle desarrollado en su superficie. Por esta razón, no se utilizó directamente como modelo final, sino como referencia para reconstruir manualmente una versión con una menor densidad de polígonos, denominada Low Poly.

Durante la retopología se procuró mantener la silueta y las proporciones principales definidas en la versión High Poly, distribuyendo la geometría de acuerdo con las características de cada zona del personaje. En áreas con menor variación superficial se utilizó una cantidad menor de polígonos, mientras que las regiones que requerían mayor definición conservaron una distribución geométrica más detallada.

También se consideró la organización de los bucles de aristas (edge loops) en aquellas zonas destinadas a experimentar deformaciones durante la animación, principalmente alrededor de articulaciones y otras áreas de movimiento. Esta distribución se realizó con el propósito de preparar la malla para las etapas posteriores de rigging y animación.

Ilustración 18

Proceso de retopología.

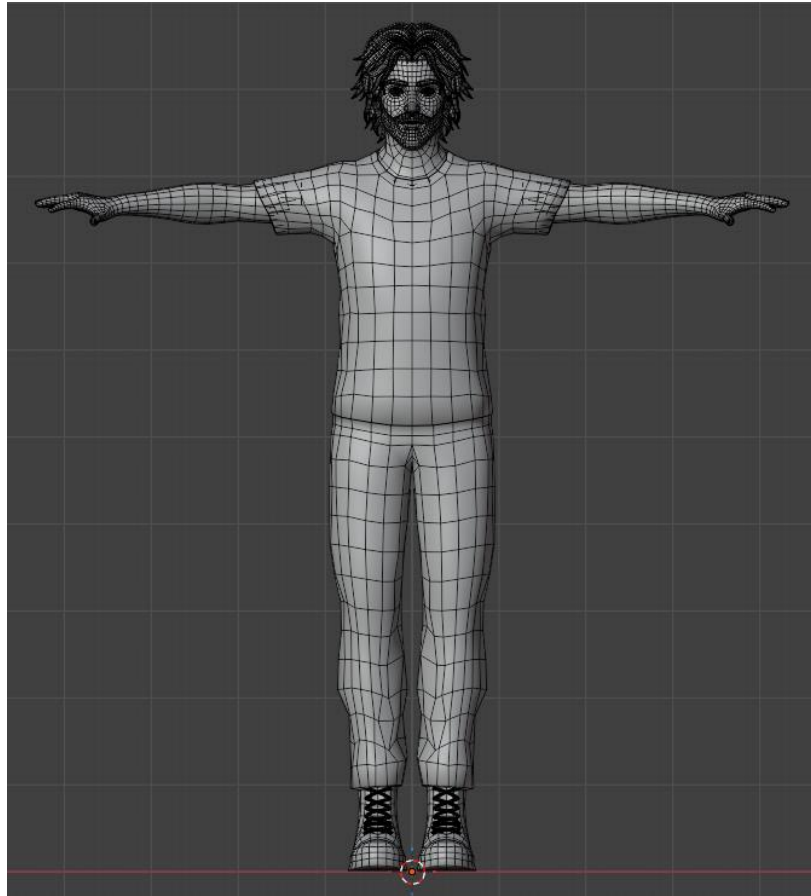


Nota. Reconstrucción de la geometría Low Poly sobre el modelo High Poly mediante herramientas de retopología en Blender. Elaboración propia.

El proceso de reconstrucción permitió diferenciar claramente dos versiones del personaje: el modelo High Poly, utilizado como referencia de detalle, y el modelo Low Poly generado mediante retopología, utilizado para continuar con el mapeado UV, baking, texturizado, rigging y las demás etapas de preparación del asset.

Ilustración 19

Modelo Low Poly finalizado.



Nota. Resultado de la retopología del personaje, preparado para las etapas posteriores del pipeline. Elaboración propia.

Como resultado de esta etapa se obtuvo una nueva malla con una menor complejidad geométrica respecto al modelo High Poly y con una topología organizada para continuar el pipeline de producción. Esta reducción corresponde a un proceso de optimización geométrica mediante retopología; su efecto específico sobre el rendimiento del videojuego deberá evaluarse posteriormente mediante datos técnicos obtenidos de los modelos y del proyecto en Unity.

4.2.7. Mapeado UV (UV Mapping)

Después de finalizar la retopología de los personajes, se realizó el mapeado UV en Blender mediante el espacio de trabajo UV Editing. Esta etapa permitió representar la superficie tridimensional del modelo sobre un espacio bidimensional para preparar la geometría utilizada posteriormente durante los procesos de baking y texturizado.

El procedimiento comenzó con la definición de costuras (seams) sobre diferentes zonas del modelo. Estas fueron ubicadas principalmente en áreas menos visibles, como la parte interna de determinadas extremidades, la espalda o regiones parcialmente cubiertas por la vestimenta, con el propósito de separar la geometría en secciones que facilitaran su despliegue.

Una vez definidas las costuras, se realizó el despliegado de la geometría, obteniendo las diferentes islas UV (UV Islands) correspondientes a las partes que conformaban el personaje. Posteriormente, estas islas fueron reorganizadas manualmente dentro del espacio UV para aprovechar el área disponible y mantener una distribución adecuada entre los diferentes componentes del modelo.

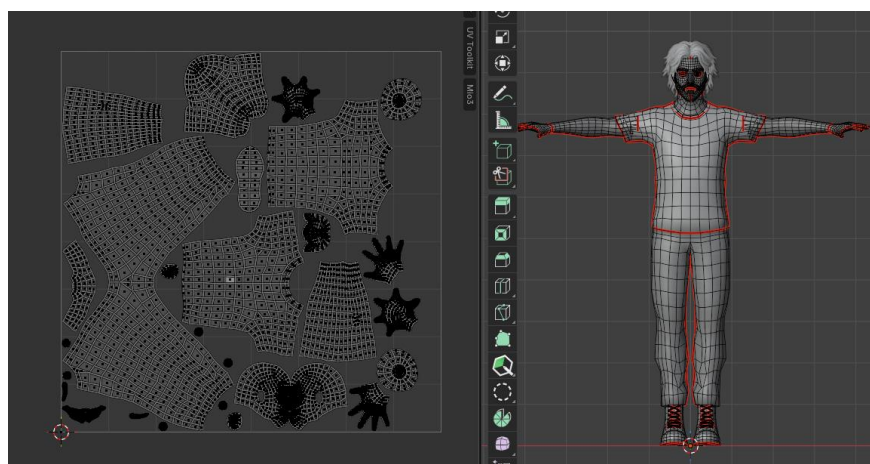
Durante esta organización se revisó visualmente la escala relativa de las distintas islas con el propósito de evitar diferencias excesivas entre zonas del personaje. También se verificó la existencia de posibles superposiciones y la distribución general de las coordenadas antes de continuar con las siguientes etapas del pipeline.

En determinados casos se reutilizaron partes simétricas del modelo mediante la superposición intencional de sus islas UV. Este procedimiento permitió compartir la misma información de textura entre geometrías simétricas y reducir la necesidad de asignar un espacio UV independiente a cada una de ellas.

La organización de las islas se realizó considerando las características de cada modelo y las necesidades posteriores de texturizado. Cuando fue necesario, se realizaron ajustes en la posición, escala y orientación de determinadas islas para mejorar su distribución dentro del espacio disponible.

Ilustración 20

Resultado final del mapa UV.



Nota. Organización de las islas UV del modelo Low Poly realizada en Blender. Elaboración propia.

Como resultado de esta etapa se obtuvieron los mapas UV correspondientes a los modelos trabajados, preparados para continuar con el proceso de baking y posterior texturizado en Substance 3D Painter.

4.2.8. Desarrollo de objetos y props tridimensionales

Además de los personajes y escenarios principales, se desarrollaron diferentes objetos tridimensionales destinados a complementar los espacios y las mecánicas presentes en los niveles de InkFall. Dentro de este grupo se incluyeron armas, trampas y otros elementos requeridos para la interacción y ambientación de los escenarios.

El desarrollo de estos recursos comenzó con la revisión de referencias visuales y de la función que cada objeto debía cumplir dentro del videojuego. A partir de esta

información se definieron sus formas generales, proporciones y características visuales antes de iniciar el proceso de modelado.

A diferencia del flujo aplicado a los personajes, en el que se trabajó inicialmente con una versión High Poly seguida de retopología, determinados props fueron contruidos directamente como modelos Low Poly en Blender. Este procedimiento se aplicó principalmente a objetos cuya geometría podía desarrollarse con una complejidad menor sin requerir un proceso previo de escultura de alta resolución.

Durante el modelado se construyeron las formas principales de cada objeto mediante técnicas de modelado poligonal, procurando mantener una geometría organizada y adecuada a las características visuales requeridas para cada recurso. La complejidad de la malla se adaptó según la forma y función del objeto, evitando incorporar geometría que no fuera necesaria para definir su silueta o sus elementos principales.

Una vez finalizada la geometría se realizó el correspondiente mapeado UV para preparar los modelos para la aplicación de materiales y texturas.

Ilustración 21

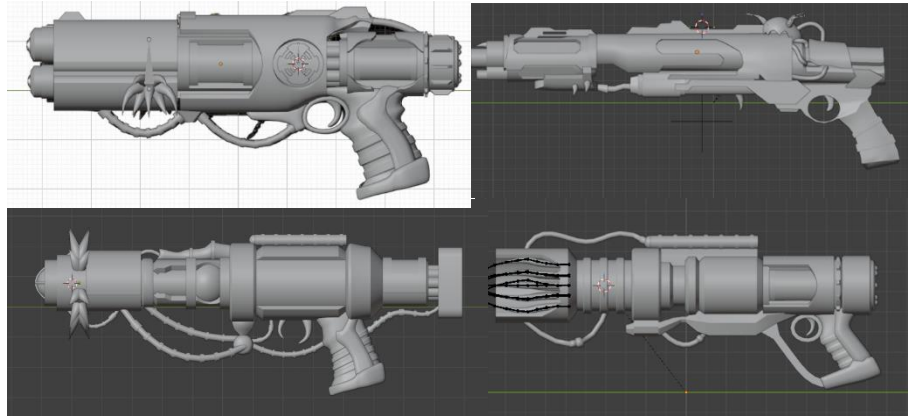
Modelado 3D de trampas del nivel Concret Loop



Nota. Proceso de modelado de las trampas desarrolladas para el nivel Concret Loop en Blender. Elaboración propia.

Ilustración 22

Modelado de armas para el primer nivel de Concret Loop



Nota. Modelado 3D de armas para el nivel Concret Loop en Blender. Elaboración propia.

Como resultado de esta etapa se obtuvo un conjunto de props tridimensionales preparados para continuar con el proceso de texturización.

4.3. Materiales y texturas

4.3.1. Generación de mapas de malla mediante Baking

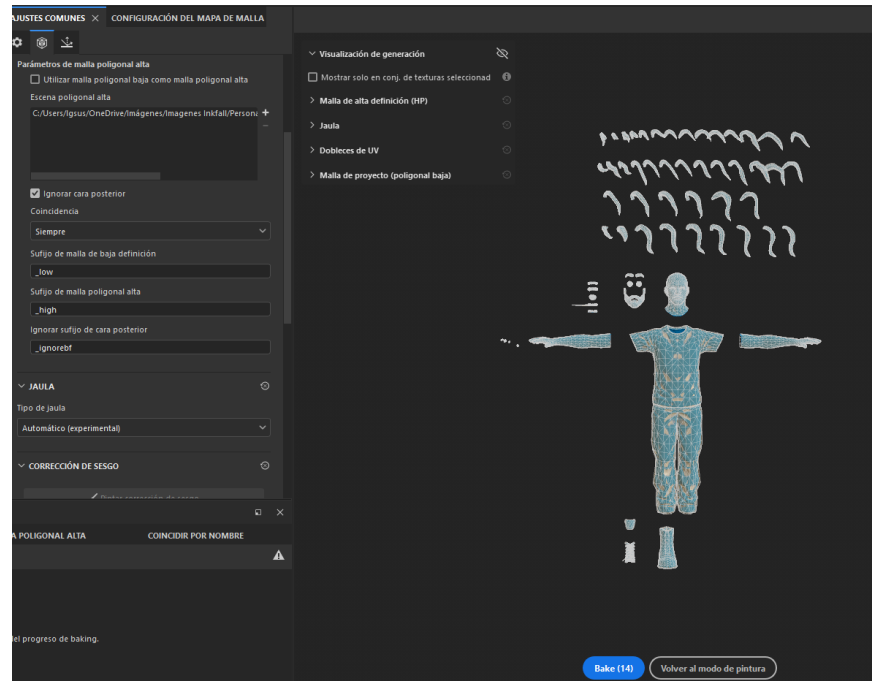
La generación de mapas de malla se realizó en Substance 3D Painter mediante la opción Generar mapas de malla, adaptando la configuración de acuerdo con las características de cada asset. El procedimiento utilizado no fue idéntico para todos los recursos, debido a que determinados modelos fueron desarrollados mediante un flujo High Poly–Low Poly, mientras que otros fueron creados directamente con una geometría de menor complejidad.

En el caso de los personajes, se utilizaron las versiones High Poly y Low Poly desarrolladas previamente en Blender. El modelo High Poly fue utilizado como referencia durante el proceso de baking para transferir información superficial hacia la versión Low Poly. A partir de este procedimiento se generaron mapas auxiliares como

Normal, Ambient Occlusion, Curvature, World Space Normal, Position y Thickness, utilizados posteriormente durante el proceso de texturizado.

Ilustración 23

Configuración para la generación de mapas de malla en Substance 3D Painter



Nota. Proceso de generación de mapas auxiliares utilizando el modelo High Poly como referencia para la versión Low Poly en Substance 3D Painter. Elaboración propia.

Tabla 6

Mapas generados durante el proceso de baking

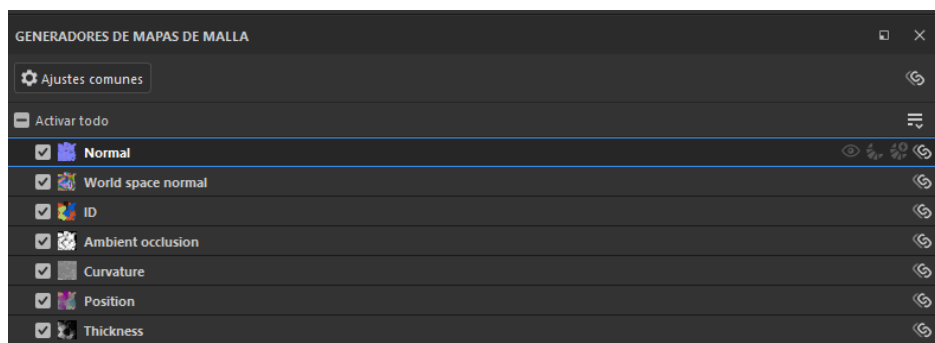
Mapa	Función dentro del proceso
Normal Map	Simula detalles geométricos del modelo High Poly sobre la superficie del modelo Low Poly, mejorando la apariencia visual sin aumentar la cantidad de polígonos.
Ambient Occlusion (AO)	Representa las zonas donde la luz tiene menor incidencia, proporcionando mayor profundidad y realismo.
Curvature	Identifica bordes y cambios de superficie para facilitar la aplicación automática de desgaste, suciedad y otros efectos durante el texturizado.
World Space Normal	Almacena la orientación espacial de las normales y sirve como referencia para diversos efectos procedurales.
Position Map	Registra la posición de cada punto del modelo dentro del espacio tridimensional, permitiendo aplicar materiales dependientes de la ubicación.
Thickness Map	Calcula el espesor del modelo para generar efectos relacionados con la transmisión de luz y variaciones de material.

Nota. Se presentan los principales mapas auxiliares generados mediante el proceso de baking en Substance 3D Painter y la función que cumple cada uno durante la preparación y texturizado de los modelos. Elaboración propia.

Una vez generados los mapas se realizó una revisión visual para identificar posibles errores de proyección o artefactos. Cuando se detectaron inconsistencias, fue necesario revisar elementos relacionados con la geometría, las coordenadas UV o la configuración del baking antes de repetir el procedimiento.

Ilustración 24

Mapas de malla generados mediante baking en Substance 3D Painter



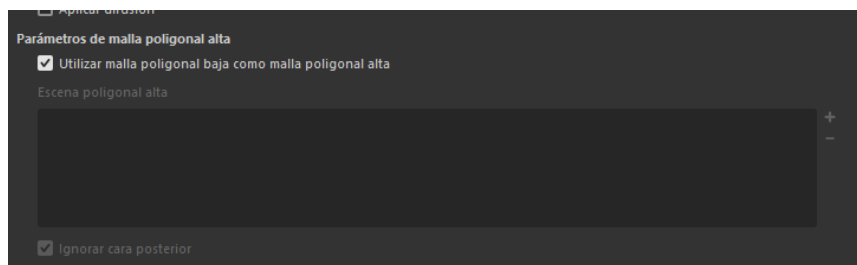
Nota. Mapas de malla generados tras hacer el bake en Substance 3D Painter. Elaboración propia.

Como resultado de esta etapa se obtuvieron los mapas auxiliares utilizados posteriormente durante el texturizado de los personajes en Substance 3D Painter. El Normal Map permitió conservar visualmente determinados detalles procedentes de la versión High Poly, mientras que los demás mapas proporcionaron información auxiliar para la aplicación de materiales, máscaras, generadores y efectos sobre la superficie del modelo.

Por otra parte, determinados recursos, como el laberinto, las armas y las trampas, fueron desarrollados directamente mediante una geometría de menor complejidad y no contaban con una versión High Poly independiente. Para estos casos, dentro de la configuración de baking de Substance 3D Painter se habilitó la opción Utilizar malla poligonal baja como malla poligonal alta, permitiendo generar los mapas de malla necesarios a partir de la propia geometría del modelo.

Ilustración 25

Opción de utilizar malla poligonal baja como malla poligonal alta



Nota. Activación de la opción “Utilizar malla poligonal baja como malla poligonal alta” para generar los mapas de malla de recursos desarrollados directamente como Low Poly. Elaboración propia.

En estos recursos, los detalles adicionales de superficie no procedieron de una transferencia geométrica desde un modelo High Poly. En su lugar, fueron incorporados posteriormente durante el proceso de texturizado mediante materiales, mapas de normales, información de altura y otros recursos disponibles en Substance 3D Painter.

Los mapas generados mediante ambos procedimientos fueron utilizados como información auxiliar durante la creación de materiales, máscaras, generadores y efectos de superficie en la etapa de texturizado.

4.3.2. Texturizado de los assets tridimensionales

Una vez realizado el mapa de mallas, se realizó el proceso de texturizado en el mismo programa de Substance 3D Painter. El procedimiento fue adaptado de acuerdo con las características de cada recurso, debido a que los personajes, objetos y elementos del escenario presentaban diferentes tipos de superficies y requerimientos visuales.

El texturizado se desarrolló mediante un flujo de trabajo basado en materiales PBR (Physically Based Rendering), trabajando propiedades como el color base, la rugosidad, el comportamiento metálico y los detalles superficiales representados mediante mapas de normales. Para organizar el proceso se utilizaron principalmente

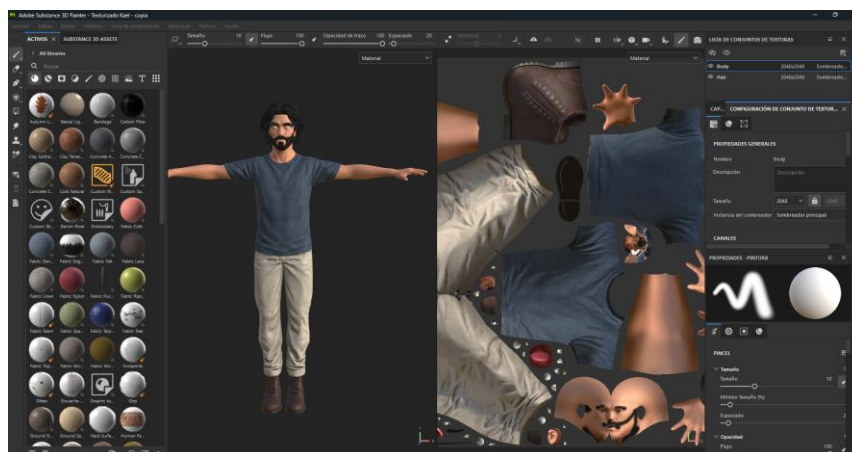
capas de relleno (Fill Layers), a las cuales se asignaron colores y propiedades de material de acuerdo con las diferentes zonas de cada asset.

Las capas de relleno fueron combinadas con máscaras para delimitar las regiones sobre las cuales se aplicaba cada material. De esta manera, fue posible trabajar de forma independiente elementos como piel, vestimenta, accesorios y diferentes superficies presentes en los modelos sin modificar directamente las demás zonas del recurso.

Además de la creación manual de materiales mediante capas de relleno, se utilizaron materiales inteligentes (Smart Materials) disponibles en Substance 3D Painter como punto de partida para representar determinados tipos de superficies. Estos materiales fueron empleados principalmente en elementos como cuero, metal y otras superficies que requerían variaciones de rugosidad, color y detalle. Sus parámetros fueron posteriormente ajustados de acuerdo con las características visuales requeridas para cada asset.

Ilustración 26

Texturización de modelo 3D

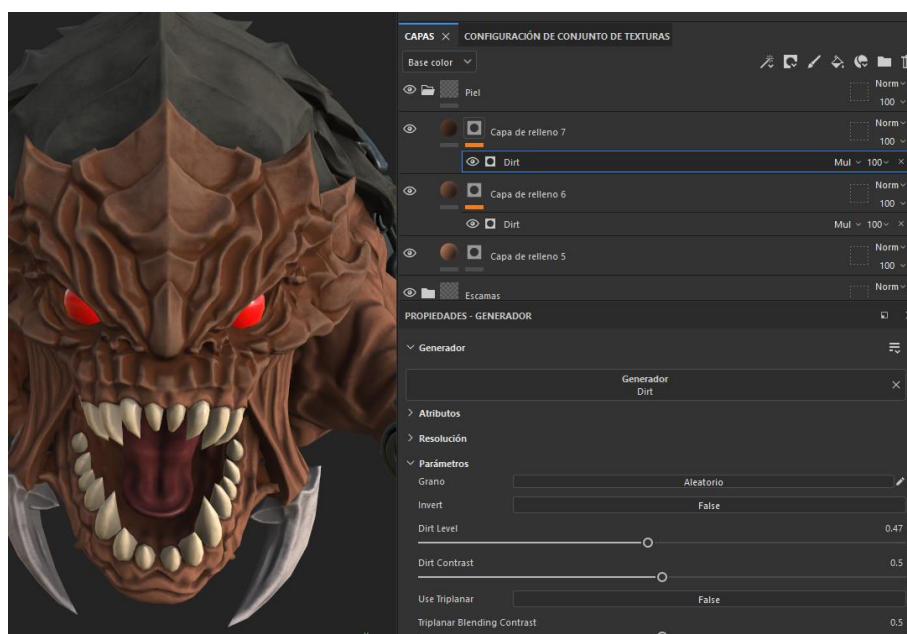


Nota. Texturización del personaje principal en Substance 3D Painter.

También se emplearon generadores como Dirt para incorporar variaciones de suciedad, desgaste y acumulación sobre determinadas superficies. Este procedimiento fue utilizado principalmente durante el texturizado de los enemigos, combinando el generador con máscaras y capas para controlar las zonas en las que aparecían estos efectos y ajustar su intensidad de acuerdo con la apariencia definida para cada personaje.

Ilustración 27

Aplicación de generadores de Dirt para el monstruo del laberinto



Nota. Uso del generador Dirt junto con máscaras para incorporar variaciones de suciedad y desgaste durante el texturizado del enemigo. Elaboración propia.

En los recursos desarrollados directamente mediante geometría de menor complejidad, como armas, trampas y determinados elementos del escenario, parte de los detalles superficiales se incorporó durante el propio proceso de texturizado. En estos casos, los relieves y variaciones visuales fueron generados mediante la información de normales y otros parámetros de los materiales empleados en Substance 3D Painter, sin

depender de la transferencia de detalles procedentes de un modelo High Poly independiente.

Ilustración 28

Texturización de armas del primer nivel



Nota. Aplicación de materiales y detalles superficiales a las armas mediante Substance 3D Painter. Elaboración propia.

Ilustración 29

Texturización de trampas del primer nivel



Nota. Aplicación de materiales y detalles superficiales a las trampas mediante Substance 3D Painter. Elaboración propia.

En el caso del laberinto de Concret Loop, el procedimiento fue diferente debido a la extensión de las superficies que conforman sus paredes. En lugar de importar y texturizar la geometría completa del escenario en Substance 3D Painter, se preparó

previamente en Blender un plano que representaba la superficie base sobre la cual se desarrollaría el material. Este plano fue exportado e importado en Substance 3D Painter para realizar sobre él el proceso de texturizado.

Sobre esta superficie se aplicaron capas de relleno, materiales y detalles visuales hasta obtener la apariencia requerida para las paredes del nivel. El propósito de trabajar sobre un plano independiente fue generar una textura que pudiera repetirse posteriormente sobre las superficies extensas del laberinto sin perder calidad.

Ilustración 30

Textura creada para el mapa del laberinto



Nota. Material texturizado a partir de un plano en Substance 3D Painter para su posterior aplicación repetitiva en las paredes del laberinto mediante el parámetro Tiling de Unity.

Una vez finalizado el trabajo de materiales, se prepararon las texturas para su exportación. Los principales mapas utilizados dentro del flujo de trabajo fueron los siguientes:

Tabla 7

Principales mapas de textura utilizados en el flujo PBR

Mapa	Función dentro del proceso
Base Color	Define el color principal del material sin efectos de iluminación.
Normal Map	Simula pequeños detalles geométricos sobre la superficie del modelo.
Roughness	Controla el nivel de rugosidad o brillo del material.
Metallic	Determina el comportamiento metálico de determinadas superficies.
Ambient Occlusion	Refuerza las sombras de contacto y aporta mayor sensación de profundidad.

Nota. Se presentan los principales mapas de textura utilizados para definir las propiedades visuales de los materiales

desarrollados para los assets tridimensionales de InkFall. Elaboración propia.

Las texturas fueron exportadas desde Substance 3D Painter como archivos de imagen PNG y organizadas junto con los recursos correspondientes a cada modelo.

Como resultado de esta etapa se obtuvieron las texturas correspondientes a los assets trabajados, preparados para su posterior configuración dentro de Unity.

4.4. Preparación y animación de personajes y objetos

4.4.1. Rigging de personajes

Una vez preparados los modelos tridimensionales de los personajes, se realizó la etapa de rigging, cuyo propósito fue incorporar estructuras óseas que permitieran controlar posteriormente sus movimientos y deformaciones durante la reproducción de animaciones. Debido a las diferencias anatómicas existentes entre los personajes de InkFall, se utilizaron dos procedimientos de rigging diferentes.

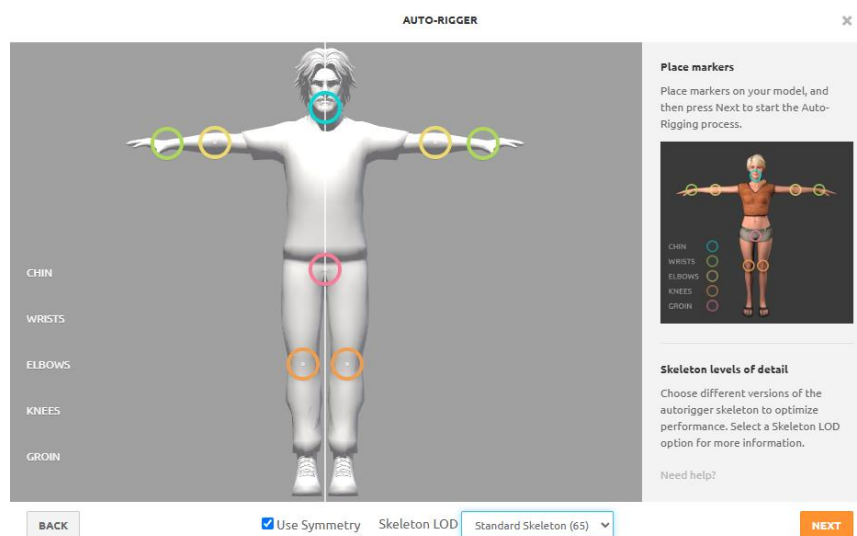
En el caso de los personajes humanoides, como el protagonista y el payaso, se utilizó la plataforma Mixamo para generar una estructura ósea mediante su sistema de

auto-rigging. Para realizar este procedimiento, los modelos fueron exportados desde Blender y cargados en la plataforma, donde se establecieron los puntos anatómicos requeridos para identificar zonas como la cabeza, brazos, codos, muñecas, torso, rodillas y pies.

A partir de estos puntos, Mixamo generó automáticamente la estructura ósea correspondiente al personaje, permitiendo disponer de un esqueleto humanoide compatible con las animaciones disponibles dentro de la misma plataforma.

Ilustración 31

Configuración del auto-rigging del personaje humanoide en Mixamo.

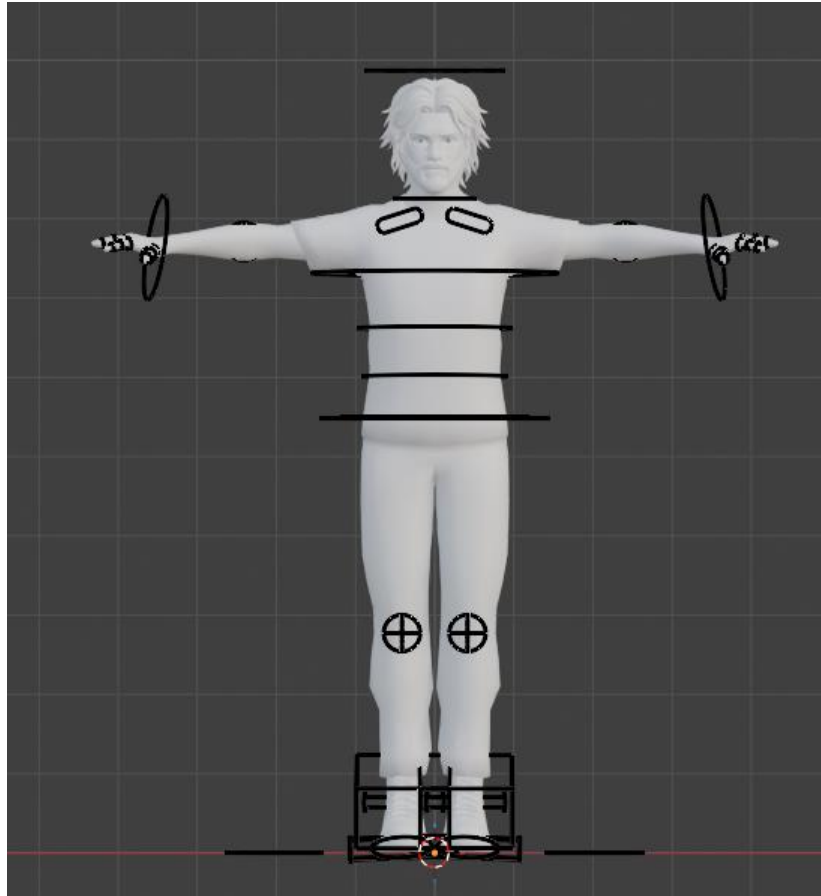


Nota. Definición de los puntos anatómicos utilizados por Mixamo para generar automáticamente el rig del personaje.

Después de generar el rig, el personaje fue importado nuevamente en Blender para revisar la estructura ósea y el comportamiento de la geometría. Durante esta etapa se realizaron ajustes en la posición de determinados huesos cuando fue necesario y se revisaron los pesos de influencia mediante Weight Paint, con el propósito de corregir deformaciones detectadas en las articulaciones.

Ilustración 32

Personaje humanoide importado en Blender con el esqueleto generado por Mixamo.



Nota. Rig del personaje humanoide importada y revisada posteriormente en Blender. Elaboración propia.

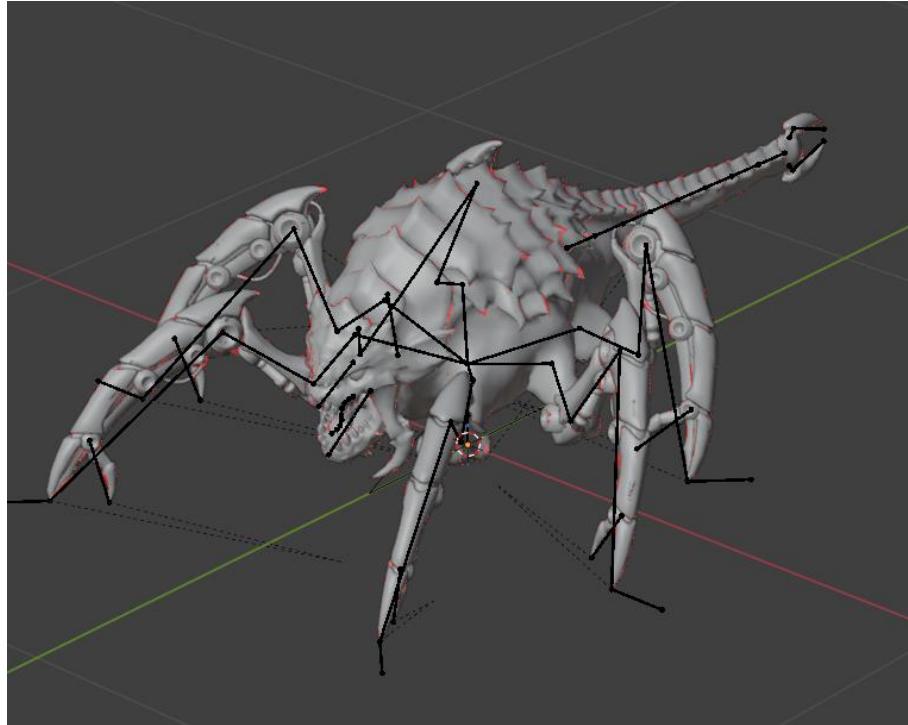
El procedimiento utilizado para el enemigo del primer nivel fue diferente. Este personaje corresponde a una criatura mecánica de seis patas y una cola similar a la de un escorpión, por lo que su anatomía no era compatible con el sistema automático de Mixamo. Por esta razón, su estructura ósea fue desarrollada manualmente en Blender.

Durante este proceso se colocaron los huesos correspondientes a las diferentes partes del modelo y se organizó su jerarquía de acuerdo con la estructura de la criatura. La disposición de los huesos tomó en consideración principalmente las patas, el cuerpo

y la cola del enemigo, debido a que estos elementos requerían control independiente durante las posteriores etapas de animación.

Ilustración 33

Sistema de rigging manual del monstruo de seis patas.

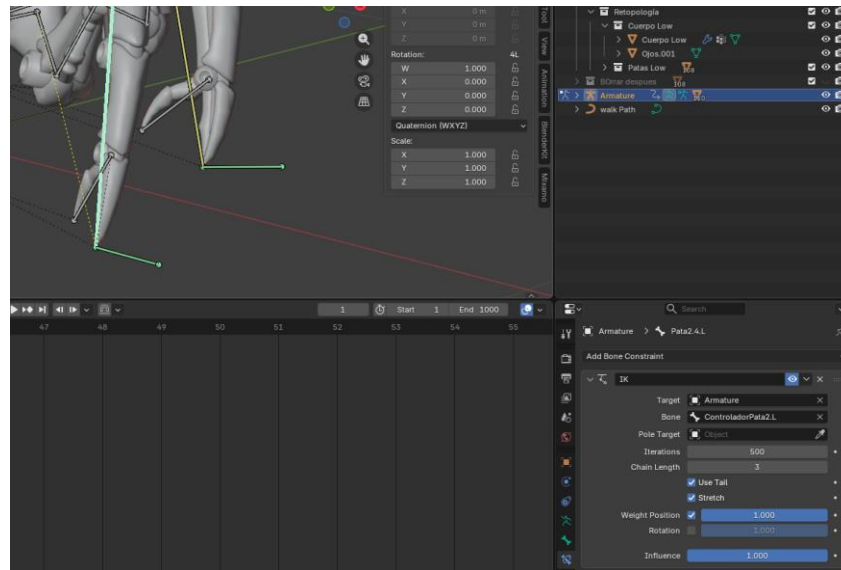


Nota. Sistema de huesos desarrollado manualmente en Blender para controlar el cuerpo, las seis extremidades y la cola del enemigo. Elaboración propia.

Para controlar el movimiento de las patas se implementó un sistema de Cinemática Inversa (Inverse Kinematics – IK) mediante restricciones y controladores en Blender. Este sistema permitió controlar la posición de las extremidades a partir de objetivos IK, en lugar de modificar individualmente cada hueso de la cadena durante la creación de determinadas animaciones.

Ilustración 34

Configuración del sistema de Cinemática Inversa (IK) en las patas del monstruo.



Nota. Controladores y restricciones IK implementados en Blender para facilitar el movimiento de las patas del personaje.

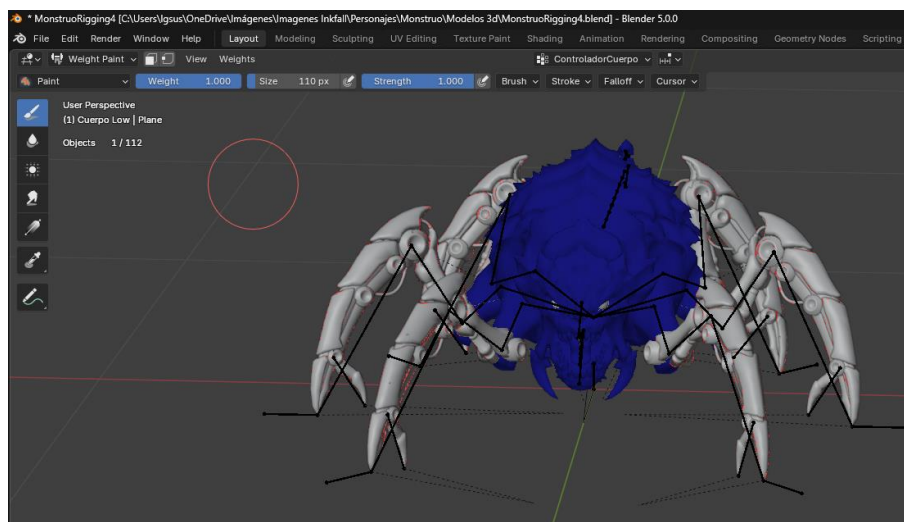
Elaboración propia.

Una vez construidas las estructuras óseas, se realizó el proceso de skinning para asociar la geometría con los huesos correspondientes. Posteriormente se revisaron los pesos de influencia mediante Weight Paint, determinando qué zonas de la malla eran afectadas por cada hueso durante las deformaciones.

Para comprobar el comportamiento del rig se realizaron poses de prueba, observando principalmente las zonas de articulación y corrigiendo los pesos cuando se detectaban deformaciones no deseadas.

Ilustración 35

Proceso de Weight Paint.



Nota. Distribución de la influencia de los huesos sobre la geometría del personaje mediante Weight Paint en Blender.

Elaboración propia.

Como resultado de esta etapa se obtuvieron los sistemas de rigging utilizados para preparar los personajes trabajados para la fase de animación. Los personajes humanoides utilizaron una estructura ósea generada mediante Mixamo y posteriormente revisada en Blender, mientras que el enemigo de seis patas contó con un rig desarrollado manualmente en Blender e incorporó controladores IK para sus extremidades.

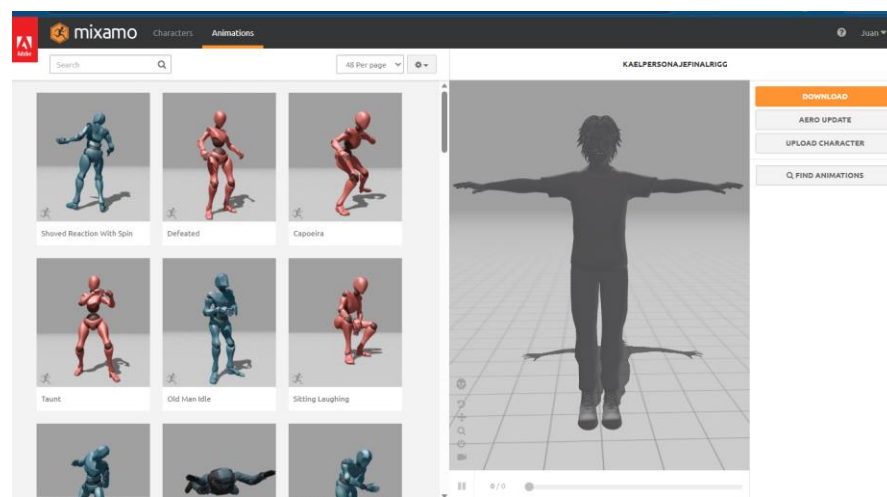
4.4.2. Animación de personajes

Una vez finalizado el proceso de rigging, se incorporaron las animaciones correspondientes a los personajes desarrollados para InkFall. Debido a las diferencias anatómicas entre los modelos, se utilizaron procedimientos distintos para los personajes humanoides y para el enemigo no humanoide del primer nivel.

En el caso de los personajes humanoides, como el protagonista y el payaso, se utilizó la biblioteca de animaciones disponible en Mixamo. Después de disponer del rig humanoide, se seleccionaron animaciones de acuerdo con las acciones requeridas dentro del videojuego, entre ellas movimientos de desplazamiento, carrera, reposo (Idle) y otras acciones relacionadas con el comportamiento de los personajes.

Ilustración 36

Selección de animaciones para los personajes humanoides en Mixamo.



Nota. Selección de animaciones base utilizadas para los personajes humanoides del prototipo mediante la biblioteca de Mixamo.

Las animaciones seleccionadas fueron descargadas desde Mixamo y posteriormente incorporadas al flujo de trabajo del proyecto. Durante su revisión se comprobó visualmente el comportamiento de la geometría en las articulaciones y se realizaron los ajustes necesarios cuando se identificaron deformaciones relacionadas con el rig o los pesos de influencia.

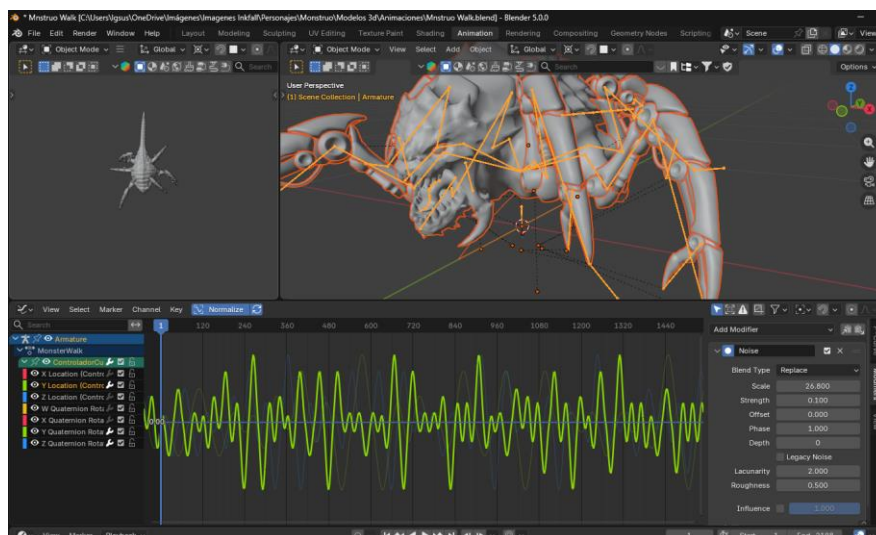
Para el enemigo no humanoide del primer nivel se utilizó un procedimiento diferente. Debido a que este personaje posee seis patas y una estructura corporal que

no corresponde a una anatomía humanoide, sus movimientos no podían obtenerse directamente mediante la biblioteca de animaciones de Mixamo.

Como primera alternativa se desarrollaron animaciones manualmente en Blender utilizando el armature construido durante la etapa de rigging. Los movimientos fueron creados mediante la manipulación de los huesos y la utilización de fotogramas clave (keyframes), apoyándose en los controladores de Cinemática Inversa previamente configurados para las extremidades.

Ilustración 37

Creación manual de la animación del monstruo de seis patas en Blender.

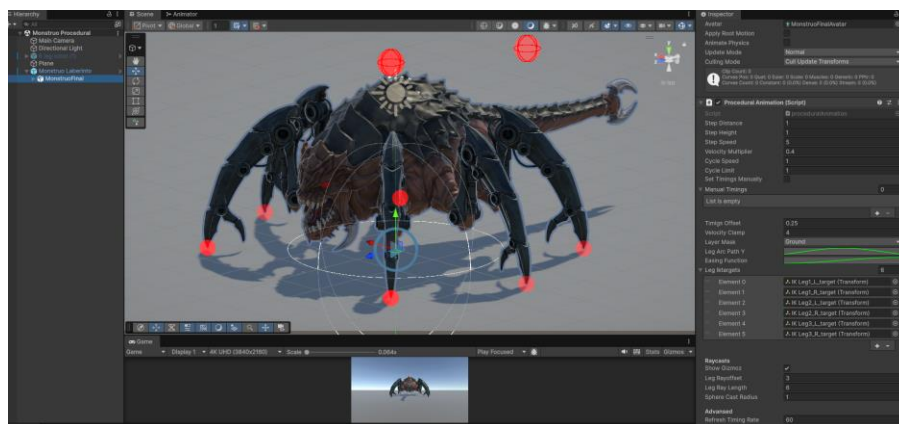


Nota. Creación de movimientos mediante fotogramas clave y controladores IK en Blender. Elaboración propia.

Adicionalmente, se realizaron pruebas con una alternativa de animación procedural mediante un asset utilizado dentro de Unity. Este procedimiento buscó generar el movimiento de las extremidades de la criatura en función de su desplazamiento, con el propósito de comparar su comportamiento con las animaciones desarrolladas manualmente.

Ilustración 38

Prueba de animación procedural del monstruo utilizando el asset de Unity.



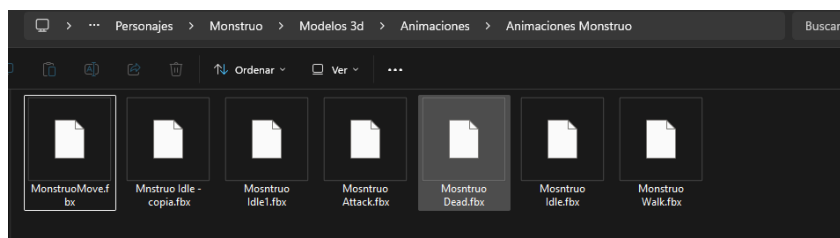
Nota. Prueba realizada mediante un sistema de animación procedural dentro de Unity para evaluar el movimiento de las extremidades del enemigo. Elaboración propia.

De esta manera, para el enemigo de seis patas se exploraron dos procedimientos: la animación manual desarrollada en Blender y una alternativa de animación procedural probada dentro de Unity. Esta comparación permitió analizar las diferencias de funcionamiento entre ambos métodos dentro del contexto del proyecto. La documentación disponible confirma la realización de las dos pruebas, por lo que cualquier selección definitiva entre ambos procedimientos deberá respaldarse con la implementación finalmente utilizada en el prototipo.

Posteriormente, las animaciones preparadas para los personajes fueron incorporadas al proyecto de Unity. Dentro del motor se organizaron los recursos de animación correspondientes para permitir su utilización junto con los modelos y sistemas de personajes.

Ilustración 39

Organización de animaciones en carpeta.



Nota. Organización de los archivos de animación correspondientes a los personajes del proyecto. Elaboración propia.

Como resultado de esta etapa se prepararon las animaciones requeridas para los personajes trabajados en el prototipo. Los personajes humanoides utilizaron recursos de animación provenientes de Mixamo, mientras que para el enemigo de seis patas se desarrollaron y probaron alternativas específicas de animación mediante Blender y Unity, adaptadas a las características de su estructura no humanoide.

4.5. Integración de recursos visuales en el motor

4.5.1. Exportación de assets

Una vez finalizadas las etapas de producción correspondientes a cada recurso tridimensional, se realizó la preparación y exportación de los archivos necesarios para su posterior incorporación al proyecto desarrollado en Unity. El procedimiento se adaptó de acuerdo con las características de cada asset, considerando si se trataba de un modelo estático, un recurso con animaciones desarrolladas directamente en Blender o un personaje humanoide cuyas animaciones provenían de Mixamo.

Los modelos tridimensionales desarrollados en Blender fueron exportados principalmente mediante el formato FBX. Este formato fue utilizado para trasladar la geometría del modelo, las coordenadas UV y, cuando correspondía, la estructura ósea y las animaciones desarrolladas directamente dentro de Blender.

Antes de generar el archivo FBX se realizó una revisión general del recurso, considerando sus transformaciones, orientación, escala y los elementos necesarios para su posterior utilización dentro del motor gráfico.

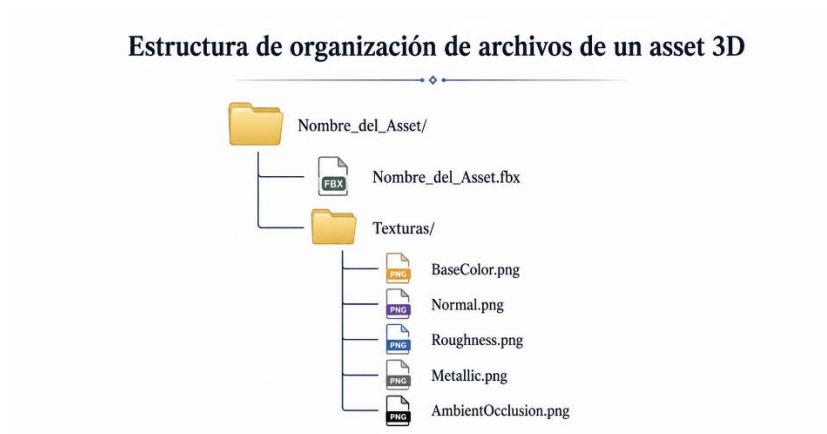
Las texturas desarrolladas en Substance 3D Painter fueron exportadas de manera independiente como archivos de imagen. Por esta razón, se mantuvieron separadas del archivo FBX principal y fueron organizadas dentro de una carpeta específica correspondiente a cada modelo para su posterior configuración en Unity.

Para mantener relacionados los diferentes archivos generados, cada asset fue almacenado dentro de una carpeta principal identificada con el nombre del modelo tridimensional. Dentro de esta carpeta se ubicó el archivo FBX correspondiente y una subcarpeta destinada a almacenar las texturas.

En los modelos estáticos y en aquellos cuyas animaciones fueron desarrolladas directamente en Blender se utilizó una estructura general similar a la siguiente:

Ilustración 40

Estructura de organización de archivos de un asset 3D



Nota. Estructura utilizada para organizar el archivo FBX y las texturas correspondientes a un asset tridimensional.

Elaboración propia.

En el caso de los recursos animados directamente en Blender, las animaciones fueron exportadas junto con el modelo dentro del mismo archivo FBX. Por este motivo, estos assets no requirieron una carpeta independiente para almacenar sus clips de animación.

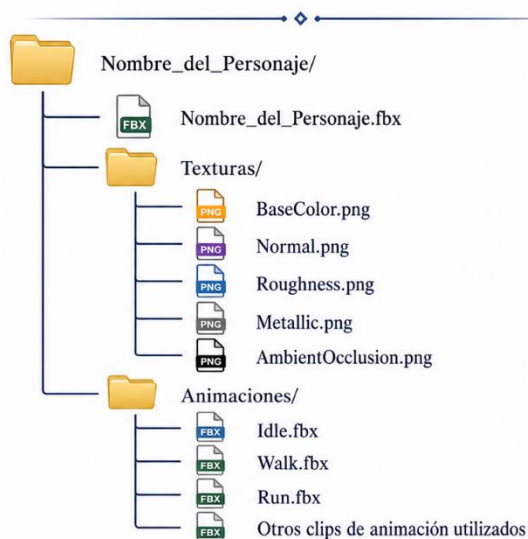
El procedimiento fue diferente para los personajes humanoides, como el protagonista y el payaso. Debido a que las animaciones utilizadas para estos personajes fueron obtenidas mediante Mixamo, los clips correspondientes fueron manejados como archivos FBX independientes del modelo principal.

Para estos personajes se empleó una organización similar a la siguiente:

Ilustración 41

Estructura de organización de archivos de un personaje humanoide

Estructura de organización de archivos de un personaje 3D



Nota. Organización del modelo, texturas y archivos de animación independientes utilizados por los personajes humanoides. Elaboración propia.

De esta manera, el archivo FBX principal de los personajes humanoides correspondió al modelo tridimensional y su estructura ósea, mientras que las

animaciones obtenidas mediante Mixamo fueron almacenadas de forma independiente dentro de la carpeta Animaciones.

Por lo tanto, la organización de los recursos dependió del procedimiento utilizado durante su producción. Los modelos cuyas animaciones fueron desarrolladas directamente en Blender conservaron estas dentro del archivo FBX principal, mientras que los personajes humanoides que utilizaron animaciones provenientes de Mixamo mantuvieron los clips como archivos independientes.

Esta estructura permitió mantener agrupados los archivos pertenecientes a cada asset antes de su incorporación al proyecto de Unity, diferenciando claramente el modelo tridimensional, sus texturas y, cuando correspondía, sus animaciones externas.

Como resultado de esta etapa se obtuvieron los recursos necesarios para continuar con la integración en Unity, organizados mediante carpetas individuales para cada asset y diferenciados según el tipo de información utilizada durante su producción.

4.5.2. Integración de los assets en Unity

Una vez preparados y organizados los archivos correspondientes a cada asset, las carpetas fueron incorporadas al proyecto desarrollado en Unity. Durante esta etapa se integraron los modelos tridimensionales en formato FBX, las texturas exportadas desde Substance 3D Painter y, cuando correspondía, los archivos independientes de animación utilizados por los personajes humanoides.

Después de importar cada recurso se realizó una revisión inicial del modelo dentro del motor gráfico. En esta comprobación se observó la geometría del asset y su disposición general dentro del proyecto, realizando los ajustes necesarios cuando se identificaban diferencias relacionadas con su escala u orientación.

Las texturas fueron importadas de manera independiente del archivo FBX. Posteriormente se configuraron los materiales correspondientes dentro de Unity y se asignaron los diferentes mapas de textura utilizados por cada asset.

La apariencia obtenida previamente en Substance 3D Painter fue utilizada como referencia durante esta configuración. Sin embargo, debido a que la visualización de un material puede variar según el shader, la iluminación y la configuración utilizada dentro del motor gráfico, los materiales fueron revisados nuevamente en Unity después de asignar sus respectivas texturas.

En el caso de los personajes y otros recursos que incluían una estructura de rigging, se revisó la información importada junto con el modelo. Cuando el recurso incorporaba animaciones desarrolladas directamente en Blender, estas se encontraban incluidas en el archivo FBX principal y podían ser revisadas después de su importación en Unity.

Para los personajes humanoides, las animaciones provenientes de Mixamo fueron importadas mediante los archivos FBX independientes almacenados previamente dentro de la carpeta Animaciones de cada personaje. Estos recursos fueron incorporados junto con el modelo principal para disponer de los clips requeridos dentro del proyecto.

Una vez incorporados el modelo, sus materiales y, cuando correspondía, las animaciones, el recurso fue preparado para su utilización dentro de las escenas del videojuego. Los assets integrados fueron posteriormente convertidos en Prefabs, permitiendo disponer de una versión reutilizable del recurso dentro del proyecto de Unity.

Ilustración 42

Integración de personajes dentro de Unity.



Nota. Recursos tridimensionales importados, configurados y preparados para su utilización dentro del prototipo de InkFall. Elaboración propia.

Finalmente, los recursos fueron utilizados dentro de los escenarios correspondientes del prototipo, donde se realizaron comprobaciones visuales de su apariencia y disposición. En los recursos animados también se revisó la reproducción de los clips disponibles según las características de cada modelo.

Como resultado de esta etapa, los modelos FBX, texturas y archivos de animación preparados durante las fases anteriores quedaron incorporados al proyecto de Unity y configurados como recursos utilizables dentro del prototipo de InkFall.

4.6. Inventario de assets desarrollados

Como resultado de la implementación de la presente propuesta tecnológica se desarrollaron diversos assets tridimensionales destinados al videojuego InkFall. Estos

recursos comprenden personajes, enemigos, el laberinto correspondiente al nivel Concret Loop y diferentes objetos interactivos, elaborados mediante el flujo de trabajo descrito en el presente manual técnico.

La **Tabla 8** presenta el inventario general de los principales assets desarrollados durante el proyecto.

Tabla 8

Inventario de assets desarrollados

Asset	Tipo	Nivel de Videojuego	Estado
Personaje Principal	Personaje	General	Finalizado
Payaso	Personaje	Nivel 2	Finalizado
Monstruo	Personaje	Nivel 1	Finalizado
Trampas	Objeto interactivo	Nivel 1 y 2	Finalizado
Armas	Objeto interactivo	Nivel 1	Finalizado
Vela	Objeto interactivo	Nivel 2	Finalizado
Soporte Arma	Objetos decorativos	Nivel 2	Finalizado
Llave	Objeto interactivo	Nivel 2	Finalizado
Escenario del laberinto	Escenario	Nivel 1	Finalizado

Nota. Se presentan los principales assets tridimensionales desarrollados para el prototipo de InkFall, clasificados de acuerdo con su tipo, nivel del videojuego y estado de desarrollo. Elaboración propia.

4.7. Validación técnica de recursos visuales

4.7.1. Criterios de validación de los assets

La validación se limitó a la comprobación técnica de los recursos tridimensionales desarrollados o preparados como parte de la propuesta y no correspondió a una evaluación de experiencia de usuario ni a una prueba integral del rendimiento del videojuego.

Para realizar la validación, cada recurso fue incorporado al proyecto de Unity utilizando los archivos preparados durante la etapa de exportación. Inicialmente se comprobó la correcta importación y visualización de la geometría. Posteriormente se

revisaron la escala y orientación del modelo dentro de la escena, así como la asignación y visualización de los materiales y mapas de textura correspondientes.

En los personajes y recursos que disponían de una estructura de rigging, se verificó que el esqueleto pudiera ser reconocido junto con la geometría y que las deformaciones necesarias para su utilización fueran funcionales. De igual manera, cuando el asset incluía animaciones, se comprobó que los clips correspondientes pudieran reproducirse dentro del proyecto.

Finalmente, se verificó la incorporación de cada recurso dentro de la escena o contexto para el cual fue desarrollado. A partir de estas comprobaciones se establecieron los criterios presentados en la **Tabla 9**.

Tabla 9

Criterios de validación de los recursos visuales

Criterio	Aspecto evaluado
Geometría	El modelo se importa y visualiza correctamente dentro de Unity.
Escala y orientación	El recurso mantiene una escala y orientación adecuadas para su utilización dentro de la escena.
Materiales y texturas	Los materiales y mapas de textura pueden ser asignados y visualizados correctamente.
Rigging	La estructura ósea es reconocida y funciona con la geometría del recurso cuando corresponde.
Animaciones	Los clips de animación pueden reproducirse correctamente cuando el asset requiere movimiento.
Integración en escena	El asset puede incorporarse y utilizarse dentro de la escena correspondiente del prototipo.

Nota. Se presentan los criterios técnicos utilizados para comprobar los assets tridimensionales integrados en Unity,

considerando geometría, escala y orientación, materiales y texturas, rigging, animaciones e integración en escena.

Elaboración propia.

4.7.2. Resultados de la validación en Unity

Una vez establecidos los criterios, se realizó la comprobación de los principales assets tridimensionales desarrollados durante el proyecto. Para registrar los resultados se utilizaron los estados Cumple, No cumple y No aplica.

El estado Cumple indica que el recurso respondió correctamente a la comprobación realizada; No cumple identifica la existencia de un inconveniente que impide satisfacer el criterio establecido; y No aplica corresponde a aquellos criterios que no forman parte de las características del recurso evaluado, como el rigging o la animación en un modelo estático.

La **Tabla 10** presenta los resultados obtenidos durante la validación de los recursos visuales.

Tabla 10

Resultados de la validación técnica de los assets

Asset	Geometría, escala y orientación	Materiales y texturas	Rigging	Animaciones	Integración en escena	Resultado general
Protagonista	Cumple	Cumple	Cumple	Cumple	Cumple	Cumple
Payaso	Cumple	Cumple	Cumple	Cumple	Cumple	Cumple
Monstruo de seis patas	Cumple	Cumple	Cumple	Cumple	Cumple	Cumple
Laberinto de <i>Concret Loop</i>	Cumple	Cumple	No aplica	No aplica	Cumple	Cumple
Armas	Cumple	Cumple	No aplica	No aplica	Cumple	Cumple
Trampas	Cumple	Cumple	No aplica	No aplica	Cumple	Cumple
Otros	Cumple	Cumple	No aplica	No aplica	Cumple	Cumple

Nota. “Cumple” indica que el recurso respondió satisfactoriamente al criterio evaluado, mientras que “No aplica”

indica que el criterio no corresponde a las características del asset.

4.8. Conclusiones

El análisis de metodologías, técnicas y herramientas relacionadas con la producción de recursos tridimensionales permitió establecer una base técnica para el desarrollo artístico de InkFall. La revisión de conceptos vinculados con modelado 3D, retopología, mapeado UV, texturizado PBR, rigging, animación, exportación e integración permitió fundamentar los procedimientos aplicados durante la producción de los assets.

La estructuración del flujo de trabajo permitió organizar las diferentes etapas de producción de acuerdo con las características de cada recurso. El pipeline comprendió procesos de planificación, modelado tridimensional, preparación geométrica, texturizado, rigging, animación, exportación e integración, considerando que no todos los assets requirieron la totalidad de estas etapas.

El desarrollo de los recursos tridimensionales permitió obtener personajes, enemigos, el laberinto correspondiente al nivel Concret Loop y diferentes objetos interactivos requeridos para el prototipo. Dependiendo de las características de cada asset, se aplicaron procesos de modelado tridimensional, escultura digital, retopología, mapeado UV, baking y texturizado. En los personajes desarrollados mediante un flujo High Poly–Low Poly, la retopología permitió obtener una versión de menor complejidad geométrica, mientras que determinados props fueron desarrollados directamente mediante geometrías de menor complejidad.

La implementación de rigging y animación permitió preparar los personajes y recursos que requerían movimiento. Los personajes humanoides utilizaron sistemas de rigging y animaciones obtenidas mediante Mixamo, mientras que el enemigo de seis

patas requirió la creación manual de una estructura ósea en Blender y la utilización de sistemas de cinemática inversa para el control de sus extremidades.

Finalmente, la exportación e integración de los assets permitió incorporar modelos, materiales, texturas, estructuras de rigging y animaciones dentro de Unity. La validación técnica permitió comprobar aspectos relacionados con geometría, escala, orientación, materiales, texturas, rigging, animaciones e integración en escena según las características de cada recurso, evidenciando la preparación de los assets desarrollados para su utilización dentro del prototipo de InkFall.

4.9. Recomendaciones

Para futuras etapas de desarrollo de InkFall, se recomienda mantener un flujo de producción estructurado para los recursos tridimensionales, conservando la separación entre archivos de trabajo, modelos FBX, texturas y animaciones externas. Esta organización facilita la localización de los recursos y permite realizar modificaciones posteriores sin alterar innecesariamente otros elementos del proyecto.

Se recomienda registrar desde el inicio de la producción las características técnicas principales de los assets, como número de vértices y triángulos, cantidad de materiales y resolución de las texturas. Disponer de esta información permitiría realizar comparaciones más precisas entre diferentes versiones de un modelo y facilitaría futuras evaluaciones relacionadas con la complejidad geométrica y el comportamiento de los recursos dentro del motor gráfico.

En los personajes y recursos animados se recomienda comprobar las deformaciones de la geometría mediante diferentes poses antes de realizar la exportación definitiva. En los personajes humanoides también resulta conveniente revisar las animaciones obtenidas mediante Mixamo después de su incorporación en

Unity, mientras que en personajes no humanoides se recomienda conservar una estructura de rigging organizada que facilite posteriores modificaciones o incorporación de nuevas animaciones.

Se recomienda documentar las versiones de software y las configuraciones de exportación e importación utilizadas. Esto permitiría mantener mayor consistencia cuando los archivos sean modificados posteriormente o utilizados por otros integrantes del equipo, especialmente en aspectos relacionados con escala, orientación, materiales, texturas, rigging y animaciones.

Cuando se utilicen modelos, materiales, animaciones u otros recursos provenientes de plataformas externas, se recomienda mantener un registro de su origen y de las condiciones de uso o licencia correspondientes. Esto permite diferenciar claramente los recursos desarrollados dentro del proyecto de aquellos obtenidos mediante herramientas como Mixamo, Tripo AI o bibliotecas externas, manteniendo una mayor trazabilidad de la contribución individual y del equipo.

Finalmente, se recomienda continuar realizando comprobaciones de los assets directamente dentro de las escenas del videojuego después de cada modificación relevante. La revisión final en Unity permite detectar diferencias de escala, orientación, materiales, texturas o animaciones que pueden no ser evidentes durante las etapas de producción realizadas en otras herramientas.

Referencias

- [1] J. Gregory, *Game Engine Architecture*, 3rd ed., Boca Raton, FL: CRC Press, 2019.
- [2] T. Akenine-Möller, E. Haines y N. Hoffman, *Real-Time Rendering*, 4th ed., A K Peters/CRC Press, 2018.
- [3] A. Manrubia Pereira, «El proceso productivo del videojuego: fases de producción,» *Historia y Comunicación Social*, vol. 19, 2014.
- [4] A. Beane, *3D Animation Essentials*, Sybex, 2012.
- [5] Blender Foundation, «Blender 4.2 Manual de Referencia,» 2024. [En línea]. [Último acceso: 15 8 2026].
- [6] Unity Technologies, *Unity User Manual*, Unity Technologies, 2017.
- [7] J. Schell, *The Art of Game Design: A Book of Lenses*, Morgan Kaufmann, 2008.
- [8] I. Sommerville, *Software Engineering*, 10th ed., Pearson, 2015.
- [9] A. Ward, *Game Character Development*, Course Technology, 2008.
- [10] F. Carrizo, «De los píxeles a la personalidad: El proceso creativo de desarrollar un personaje en 3D,» *Universidad Siglo 21*, 2023.
- [11] M. Pharr y G. Humphreys, *Physically Based Rendering: From Theory to Implementation*, Burlington: Elsevier, 2010.
- [12] E. Lengyel, *Mathematics for 3D Game Programming and Computer Graphics*, 3rd ed., Cengage Learning, 2011.
- [13] T. Mullen, *Introducing Character Animation with Blender*, Wiley Publishing, 2007.
- [14] C. F. C. M. d. P. B. L. Roberto Hernández Sampieri, *Metodología de la Investigación*, Ciudad de México, México: McGraw-Hill Education, 2018.
- [15] S. Z. a. J. Zelenova, «A Systematic Pipeline for Game-Ready 3D Assets,» *Bulletin of Science and Practice*, vol. 11, nº 7, p. 128–134, 2025.
- [16] D.-M. Lee, «Blender 3D as a Catalyst for Indie Game Development,» *International Journal of Contents*, vol. 21, nº 2, p. 67–74, 2025.
- [17] J. Wu, Z. Lou, J. Liu, D. Du, C. Guo y S. Guo, «From Visual Synthesis to Interactive Worlds: Toward Production-Ready 3D Asset Generation,» 2026.
- [18] A. Parab, N. Rathod, T. Patil, K. Deshpande y N. Deshmukh, «A 3D Storyline Using Unity Game Engine,» de *2022 2nd International Conference on Intelligent Technologies (CONIT)*, Hubli, India, 2022.
- [19] A. Begemann y J. Hutson, «Prompted Props, Human Pipelines: Evaluating AI-Generated 3D Assets for Game-Ready Environments,» *International Journal of Human Research and Social Science Studies*, vol. 3, nº 6, p. 647–661, 2026.
- [20] D. Luebke, *Level of detail for 3D graphics*, Morgan Kaufmann, 2003.
- [21] H. M. Chandler, *The Game Production Handbook*, Burlington, Massachusetts: Jones & Bartlett Learning, 2013.
- [22] A. Gosálbez Ibarra, «Diseño y creación de un personaje 3D,» *Universidad de Alicante*, 2024.
- [23] G. Ojanperä Arratia, «Character Animation in Blender,» *Oulu University of Applied Sciences*, 2025.
- [24] J. Osipa, *Stop Staring. Facial Modeling and Animation Done Right*, 3rd ed., 2010, 2010.