



**INSTITUTO SUPERIOR
TECNOLÓGICO QUITO**
Formamos tu PROPÓSITO DE VIDA

Guía práctico experimental de

DISEÑO DE VIDEOJUEGOS

Autor: Alexis Xavier Urgíles Pillalaza

Am



2023

GUÍA PRÁCTICO EXPERIMENTAL DE DISEÑO DE VIDEOJUEGOS

CARRERA: ANIMACIÓN MULTIMEDIA

PERIODO ACADÉMICO: SEPTIEMBRE 2025 – FEBRERO 2026

DOCENTE TUTOR: ALEXIS URGILÉS PILLALAZA

EDICIÓN: PRIMERA

AÑO: 2025

TRABAJO EN EDICIÓN:



EQUIPO EDITORIAL:

MSc. KEYERMAN TOAPANTA CISNEROS

MSc. DAVID ALEJANDRO PEREIRA SALAZAR

ISBN:

QUITO – ECUADOR





1.	GUÍA PRÁCTICA EXPERIMENTAL DE DISEÑO DE VIDEOJUEGOS.....	3
1.1.	Descripción de la asignatura	3
1.2.	Objetivos de la asignatura	3
1.3.	Contenidos mínimos	3
1.4.	Resultados de aprendizaje	4
2.	UNIDAD 1: Bases del diseño de Videojuegos.....	5
2.1.	Practica experimental 1	5
2.2.	Practica experimental 2	9
2.3.	Practica experimental 3	17
2	UNIDAD 2: Planificación de Proyectos.....	25
2.1	Practica experimental 4	25
2.2	Practica experimental 5	31
2.3	Practica experimental 6	34
3	UNIDAD 3: Rigging.....	37
3.1	Practica experimental 7	37
3.2	Practica experimental 8	41
3.3	Practica experimental 9	47
4	UNIDAD 4: Programación en Unreal Engine	50
4.1	Practica experimental 10	50
4.2	Practica experimental 11	56
5	UNIDAD 5: Mecánicas y Teaser	59
5.1	Practica experimental 12	59
5.2	Practica experimental 13	63



1. GUÍA PRÁCTICA EXPERIMENTAL DE DISEÑO DE VIDEOJUEGOS

1.1. Descripción de la asignatura

La asignatura Diseño de Videojuegos ofrece al estudiante una introducción práctica y conceptual al desarrollo de videojuegos, enfocándose en la creación de experiencias interactivas adaptables a múltiples plataformas. A lo largo del curso se exploran las fases del diseño, la lógica de programación, la estructura visual, la narrativa interactiva y la experiencia del jugador, integrando aspectos clave como el gameplay, el diseño de interfaces y la jerarquía de objetos. El estudiante aprende a utilizar lenguajes como HTML5 y CSS3 como base para la construcción de proyectos funcionales y personalizables.

En un mundo donde los videojuegos trascienden el entretenimiento y se consolidan como medios de comunicación, expresión y aprendizaje, esta asignatura permite al estudiante diseñar con criterio técnico, estético y narrativo. Se promueve la comprensión del flujo de trabajo en la industria del videojuego, así como el uso de recursos gráficos como bitmap sequences, pathfinding y animaciones por frames, orientados al diseño de experiencias inmersivas y significativas.

Como parte de la formación en la carrera de Animación Multimedia, esta materia potencia la capacidad del estudiante para integrar diseño visual, pensamiento lógico y narrativa interactiva, desarrollando videojuegos con sentido, funcionalidad y adaptabilidad multiplataforma. Además, aporta competencias esenciales para la inserción en industrias creativas digitales, el trabajo colaborativo y el diseño de productos orientados al usuario.

1.2. Objetivos de la asignatura

Gestionar piezas sonoras a partir de una idea creativa, mediante procesos de preproducción, grabación, edición, mezcla y masterización, para diseñar recursos auditivos funcionales y expresivos que respondan a las demandas de proyectos audiovisuales, animaciones digitales y plataformas multimedia.

1.3. Contenidos mínimos

1. Grabación de audio y formatos.
2. Edición y premezcla.
3. Ecualización.
4. Masterización.
5. Programas para creación sonora.

6. Música para videojuegos

7. Efectos de sonido.

1.4. Resultados de aprendizaje

Gestiona piezas sonoras a partir de una idea, desde la preproducción y producción hasta su grabación y mezcla, con el objetivo de su utilización según las demandas.



2. UNIDAD 1: Bases del diseño de Videojuegos

2.1. Practica experimental 1

Cargo:	Docente Especialista		
Nombre:	Alexis Urgilés		
Asignatura:	Motores de Videojuegos		
Carrera:	Alexis Urgilés	Nivel:	Tercer Nivel
Estudiante:			

ACTIVIDAD PRÁCTICO EXPERIMENTAL EN EL ENTORNO ACADÉMICO

Implementación de configuraciones especiales para crear un proyecto básico en Construct 2

Objetivos

Aplicar configuraciones iniciales en Construct 2 para preparar un proyecto base optimizado para diseño de videojuegos 2D, comprendiendo la estructura del entorno y su organización de trabajo

Antecedentes/Escenario

Construct 2 permite desarrollar videojuegos mediante un sistema visual basado en eventos. Para iniciar un proyecto sólido es necesario configurar parámetros esenciales como layout, tamaño del canvas, capas, comportamientos básicos y ajustes de optimización. Esto asegura un flujo de trabajo ordenado antes de la producción de mecánicas.

Recursos necesarios

Construct 2 instalado.

Plantilla base o proyecto vacío.

Carpeta de recursos local.

Planteamiento del problema

El diseño de videojuegos requiere iniciar con un proyecto correctamente estructurado. La ausencia de configuraciones iniciales claras puede generar problemas de rendimiento y desorden en el desarrollo.

Pasos por realizar

Paso 1: Crear un proyecto vacío y nombrarlo según la práctica.

Paso 2: Configurar la resolución del proyecto y del layout principal.

Paso 3: Crear un layer para fondo, uno para elementos interactivos y otro para interfaz.

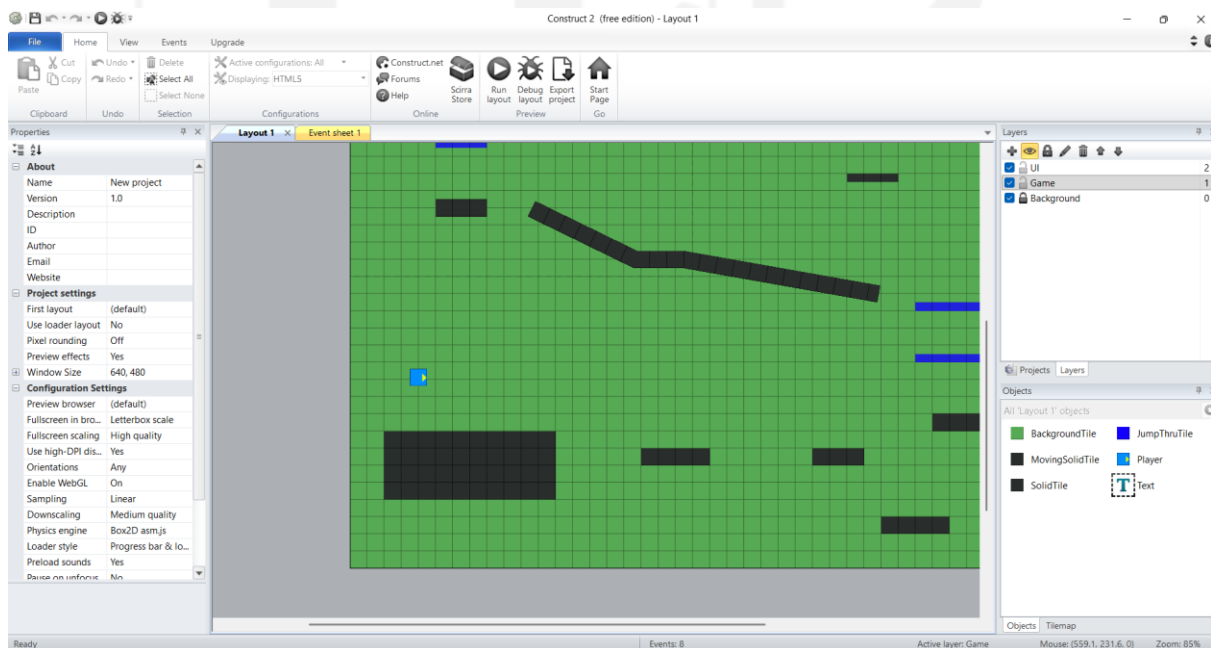
Paso 4: Configurar el tamaño del layout y ajustar la zona de juego.

Paso 5: Crear un objeto Sprite base para pruebas y asignar comportamientos iniciales.

Paso 6: Guardar el proyecto con estructura organizada de carpetas.

Desarrollo

Se crea un proyecto nuevo y se asigna un nombre representativo. Se accede a las propiedades del proyecto para ajustar la resolución general. Se abre el layout principal y se modifica su tamaño para adecuarlo a la experiencia deseada. Se crean tres capas: una destinada al fondo, otra para la jugabilidad y una capa superior para la interfaz del usuario. Se añade un sprite simple para probar interacción y se le asigna un comportamiento básico como movimiento o plataforma. Finalmente, se organiza la estructura del proyecto en carpetas para objetos, sonidos y gráficos.





Properties	
Layout properties	
Name	Layout 1
Event sheet	Event sheet 1
Active layer	Game
Unbounded scro...	No
Layout Size	1280, 1024
Margins	500, 500
Effects	
Add / edit	Effects
Project Properties	View
More information	Help

Properties	
Add / edit Instance variables	
Add / edit Behaviors	
Add / edit Effects	
Angle	0
Blend mode	Normal
Global	No
Hotspot	Top-left
Image	Edit
Initial state	Enabled
Initial visibility	Visible
Layer	Game
More information	Help
Name	SolidTile
No container	Create
Opacity	100
Plugin	Tiled Background
Position	64, 768
Size	320, 128
UID	2



Use the down arrow to allow the player to deliberately drop down through a jump-thru platform.		
1	Keyboard On Down arrow pressed	Fall Platform down through jump-thru Add action
Allow WASD as alternate controls to arrow keys.		
2	Keyboard W is down	Simulate Platform pressing Jump Add action
3	Keyboard A is down	Simulate Platform pressing Left Add action
4	Keyboard S is down	Fall Platform down through jump-thru Add action
5	Keyboard D is down	Simulate Platform pressing Right Add action
Mirror the player's image so they appear facing the right way when moving left or right.		
6	Keyboard On Left arrow pressed	Set Mirrored Add action
	- or - Keyboard On A pressed	
7	Keyboard On Right arrow pressed	Set Not mirrored Add action
	- or - Keyboard On D pressed	
If the player falls off the bottom of the layout, restart the level.		
8	Player Y > LayoutHeight	Restart layout Add action

Resultados esperados

Al finalizar la actividad, obtendrás un proyecto base optimizado y funcional en Construct 2. Lograrás una estructura organizada mediante capas diferenciadas (fondo, juego e interfaz) y una resolución de pantalla correctamente configurada. Habrás validado el entorno mediante un sprite con comportamientos activos, garantizando un flujo de trabajo profesional que previene errores de rendimiento y facilita la implementación futura de mecánicas complejas y recursos multimedia.

2.2. Practica experimental 2

Cargo:	Docente Especialista		
Nombre:	Alexis Urgilés		
Asignatura:	Diseño de Videojuegos		
Carrera:	Alexis Urgilés	Nivel:	Tercer Nivel
Estudiante:			

ACTIVIDAD PRÁCTICO EXPERIMENTAL EN EL ENTORNO ACADÉMICO

Análisis de layouts en videojuegos de distintos géneros mediante capturas y comentarios críticos

Objetivos

Analizar layouts de videojuegos de géneros variados identificando decisiones de composición visual, jerarquía, navegación y distribución de elementos.

Antecedentes/Escenario

El layout en diseño de videojuegos determina cómo el jugador percibe la información en pantalla. La organización del espacio influye en claridad, navegación, sensación de profundidad y experiencia general. Comprender su estructura permite fundamentar decisiones de diseño para futuros proyectos.

Recursos necesarios

Herramienta de capturas.

Juegos de referencia.

Editor de texto o documento gráfico.

Planteamiento del problema

Se requiere interpretar y analizar el layout de videojuegos profesionales para identificar patrones visuales efectivos y comprender cómo se aplican según el género.

Pasos por realizar

Paso 1: Seleccionar tres videojuegos de géneros diferentes.

Paso 2: Realizar capturas de pantalla de escenas donde el layout sea evidente.

Paso 3: Identificar la jerarquía visual, ubicación de UI y composición general.

Paso 4: Escribir comentarios críticos por cada captura.

Paso 5: Organizar las capturas y análisis de forma comparativa.

Desarrollo

En esta actividad se seleccionan videojuegos de distintos géneros con el fin de analizar cómo resuelven la organización visual de la información en pantalla. A partir de capturas tomadas en momentos donde el layout sea claramente identificable, se estudian la jerarquía visual, la distribución de la interfaz y la relación entre los elementos jugables y los informativos. El análisis no se limita a describir lo visible, sino que busca interpretar por qué cada elemento ocupa determinado espacio y cómo influye en la experiencia del jugador según el género del juego.

En el caso de Teamfight Tactics, perteneciente al género estrategia auto battler, el layout está diseñado para permitir una lectura simultánea de múltiples variables estratégicas. La parte central de la pantalla está dominada por el tablero hexagonal, que funciona como núcleo visual y operativo, ya que allí se desarrollan los combates automáticos. Esta ubicación central refuerza la prioridad de la acción táctica y facilita la percepción de distancias, posiciones y sinergias entre unidades. En la zona inferior se ubica el banco de campeones y los controles económicos, como la compra de unidades y la gestión del oro, estableciendo una relación directa entre planificación y ejecución. La parte superior muestra información contextual de la ronda, oponente y temporizador, elementos que orientan la toma de decisiones bajo presión de tiempo. A los costados se distribuyen paneles complementarios como inventario y sinergias activas, organizados lateralmente para no interferir con el campo principal de combate. Este diseño responde a la necesidad de claridad estratégica, permitiendo que el jugador procese información compleja sin perder el foco central del enfrentamiento.



La jerarquía visual dentro de Teamfight Tactics está organizada en función de la prioridad estratégica que el jugador necesita durante cada fase de la partida. El tablero central constituye el núcleo dominante de la pantalla, ocupando aproximadamente el sesenta por ciento del espacio visible, lo que garantiza una lectura clara del posicionamiento de las unidades, elemento fundamental en un auto battler donde la ubicación determina el resultado del combate. En segundo nivel de importancia se encuentra la tienda en la zona inferior, directamente vinculada con la economía y la construcción de la composición del equipo, aspectos que influyen en la estrategia a mediano plazo. Las sinergias, ubicadas lateralmente, permiten monitorear el estado del equipo sin interrumpir la vista

principal, mientras que la información superior cumple una función macro al mostrar fase, rival y temporizador, aportando contexto general sin competir visualmente con el área táctica.

Desde una perspectiva crítica, el layout del juego representa un equilibrio eficiente entre densidad informativa y claridad visual. A pesar de contar con un HUD cargado de datos, la organización respeta la secuencia natural de toma de decisiones: primero observar el tablero, luego revisar economía y finalmente ajustar sinergias. El uso del color refuerza esta jerarquía, ya que los campeones en la tienda presentan marcos que indican su rareza, las sinergias activas modifican su tonalidad al mejorar y el tablero mantiene una base cromática relativamente neutra para evitar distracciones. Esta distribución permite que el jugador procese simultáneamente información micro y macro sin experimentar saturación visual, favoreciendo una experiencia estratégica fluida y estructurada.



En Hollow Knight, perteneciente al género metroidvania de plataformas, el layout se caracteriza por una apuesta clara hacia el minimalismo y la inmersión ambiental. A diferencia de otros juegos con interfaces cargadas de información, aquí el HUD se reduce a los elementos estrictamente necesarios: la barra de vida representada mediante máscaras en la esquina superior izquierda y el indicador de alma ubicado junto a ella. El mapa y otros datos adicionales no permanecen visibles de forma constante, sino que se consultan desde el menú, lo que libera casi la totalidad del espacio visual para el escenario y la acción. Aproximadamente el noventa y cinco por ciento de la pantalla está dedicado al personaje, los enemigos y el entorno, reforzando la sensación de exploración y profundidad del mundo subterráneo.

La jerarquía visual responde a una lógica centrada en la exploración más que en la lectura de estadísticas. La prioridad principal es el escenario y los enemigos, ya que el jugador debe interpretar patrones de ataque, plataformas, trampas y rutas ocultas. En segundo nivel se encuentra la salud, visible pero discreta, permitiendo monitorear el estado del personaje sin distraer la atención del entorno. Finalmente, el indicador de alma ocupa un tercer nivel jerárquico, funcionando como recurso complementario para habilidades especiales. Este enfoque demuestra cómo el layout puede adaptarse al género: en lugar de saturar la pantalla con datos, el diseño potencia la atmósfera, la tensión y la observación detallada del espacio, elementos esenciales en una experiencia metroidvania.





Desde una perspectiva crítica, la decisión minimalista adoptada en Hollow Knight evita distracciones innecesarias y respeta el ritmo introspectivo que caracteriza su propuesta jugable. La estética dibujada a mano y la atmósfera melancólica del entorno exigen que la interfaz no compita visualmente con el arte, sino que lo acompañe con discreción. El jugador recibe únicamente la información indispensable para la supervivencia, lo que fortalece la inmersión y la conexión emocional con el mundo. En contraste con propuestas estratégicas más cargadas de datos como Teamfight Tactics, aquí la interfaz prácticamente se oculta para permitir que la exploración y la ambientación dominen la experiencia.

En el caso de Call of Duty: Warzone, perteneciente al género shooter táctico, el layout responde a una necesidad completamente distinta: comunicar gran cantidad de información en tiempo real sin obstaculizar la visión del entorno, que es vital para la supervivencia. En el centro se encuentra la retícula, eje absoluto de la acción inmediata y punto de enfoque constante del jugador. En la zona inferior se dispone la información de munición, recarga y tipo de arma equipada, elementos críticos que deben consultarse de forma rápida durante los enfrentamientos. En la esquina superior izquierda se ubica el minimapa, herramienta clave para la orientación espacial y detección de amenazas cercanas, mientras que en la parte superior derecha aparece el contador de jugadores restantes, reforzando la tensión propia del modo battle royale. En la zona inferior izquierda se muestra el estado del escuadrón, permitiendo coordinar estrategias en equipo. Los mensajes emergentes, como notificaciones o alertas, se sitúan en áreas periféricas para no interferir con la mira ni con la identificación de enemigos.

La jerarquía visual prioriza claramente la acción inmediata: primero la mira y los enemigos, luego la munición disponible, después el minimapa y finalmente el estado del escuadrón. Esta

organización responde a la velocidad del género, donde las decisiones se toman en fracciones de segundo y cualquier distracción puede significar la eliminación del jugador. A diferencia del enfoque contemplativo de un metroidvania, aquí el layout está diseñado para maximizar reacción, precisión y conciencia situacional constante.



Desde un análisis crítico, el HUD de Call of Duty: Warzone se percibe como altamente funcional y optimizado para la velocidad de reacción que exige el género shooter táctico. La distribución concentra casi todos los elementos informativos en los bordes de la pantalla, liberando el centro para la retícula y la identificación de objetivos, lo que demuestra una clara prioridad en la precisión del disparo y la lectura del entorno inmediato. La tipografía utilizada es grande para facilitar la lectura en situaciones de tensión, pero mantiene una composición compacta que evita invadir el espacio visual. La paleta cromática recurre a contrastes intensos y alertas visuales claras, como cambios de color cuando la munición es baja, lo que activa respuestas rápidas casi automáticas en el jugador. En contraste con la propuesta minimalista de Hollow Knight, donde la interfaz busca pasar desapercibida para favorecer la inmersión, aquí la UI es abundante pero eminentemente operacional, diseñada para asistir de manera constante y directa en la toma de decisiones bajo presión.

Comparación general entre los tres layouts

Elemento	TFT (Auto Battler)	Hollow Knight (Metroidvania)	Warzone (Shooter)
Densidad de UI	Muy alta	Muy baja	Alta
Función del layout	Estrategia y planificación	Exploración inmersiva	Acción inmediata
Ubicación crítica	Tablero central	Entorno y personaje	Centro libre (mira)
Estilo visual	Interfaces ilustradas y paneles	HUD minimalista	HUD técnico
Jerarquía	Tablero > tienda > sinergias	Entorno > vida	Mira > munición

Resultados esperados

Como resultado del análisis, se ha identificado con precisión los patrones de composición y jerarquía visual que definen a cada género estudiado. Se ha comprendido cómo el minimalismo en Hollow Knight prioriza la inmersión, mientras que la densidad informativa en TFT y Warzone optimiza la toma de decisiones tácticas. Finalmente, se ha consolidado un criterio técnico para distribuir elementos en pantalla, logrando un equilibrio eficiente entre la carga cognitiva del usuario y la funcionalidad del diseño.

2.3. Practica experimental 3

Cargo:	Docente Especialista		
Nombre:	Alexis Urgilés		
Asignatura:	Diseño de Videojuegos		
Carrera:	Alexis Urgilés	Nivel:	Tercer Nivel
Estudiante:			

ACTIVIDAD PRÁCTICO EXPERIMENTAL EN EL ENTORNO ACADÉMICO

Planificación de experiencias de videojuego según el libro The Art of Game Design

Objetivos

Objetivos

Planificar experiencias de juego utilizando principios y lentes del diseño planteados en The Art of Game Design, aplicándolos a un concepto propio de videojuego.

Antecedentes/Escenario

El diseño de videojuegos requiere considerar emociones, mecánicas, dinámicas, narrativa, estética, objetivos y retroalimentación. El libro propone una serie de lentes que ayudan a evaluar y mejorar decisiones de diseño.

Recursos necesarios

Lectura de capítulos clave.

Cuaderno de notas o documento digital.

Planteamiento del problema

Se busca fundamentar la experiencia del videojuego mediante el uso de lentes de diseño que permitan analizar su coherencia y valor para el jugador.

Pasos por realizar

Paso 1: Seleccionar el concepto de videojuego.

Paso 2: Escoger un conjunto de lentes relevantes.



Paso 3: Aplicar cada lente al proyecto respondiendo preguntas clave.

Paso 4: Redactar un documento con conclusiones y mejoras.

Desarrollo

En la fase de planificación se parte de un concepto de videojuego inspirado en la estructura atmosférica y de exploración presente en Hollow Knight, planteando una experiencia de plataformas 2D con enfoque metroidvania y narrativa ambiental. El jugador desciende a un mundo subterráneo abandonado donde la historia no se presenta mediante cinemáticas extensas, sino a través del entorno, la arquitectura en ruinas, los enemigos y fragmentos narrativos dispersos. El concepto prioriza la exploración libre, la atmósfera melancólica, la precisión en los controles y una progresión basada en habilidades que desbloquean nuevas rutas, reforzando la sensación de descubrimiento gradual.

Para estructurar la experiencia se toman como referencia los principios expuestos por Jesse Schell en *The Art of Game Design: A Book of Lenses*, seleccionando lentes que permitan evaluar el proyecto de manera integral. Estas lentes no funcionan como reglas rígidas, sino como marcos de análisis que obligan al diseñador a formular preguntas críticas sobre coherencia, emoción, reto y espacialidad.

Al aplicar la Lente del Jugador, el análisis se centra en comprender profundamente a quién va dirigida la experiencia. El perfil principal corresponde a jugadores exploradores que valoran la ambientación, la narrativa implícita y los desafíos técnicos progresivos. Se busca provocar emociones específicas como soledad, curiosidad constante, tensión ante lo desconocido y satisfacción intensa tras superar obstáculos complejos o derrotar enemigos exigentes. El diseño contempla que el jugador puede llegar con experiencia previa en plataformas, pero no necesariamente en metroidvanias complejos, por lo que las primeras áreas deben funcionar como espacios de aprendizaje implícito donde las mecánicas se enseñan a través del entorno y no mediante tutoriales invasivos. También se considera la importancia del “refugio” como espacio seguro que reduce la tensión acumulada, generando ciclos emocionales de riesgo y alivio que fortalecen la conexión con el mundo del juego.

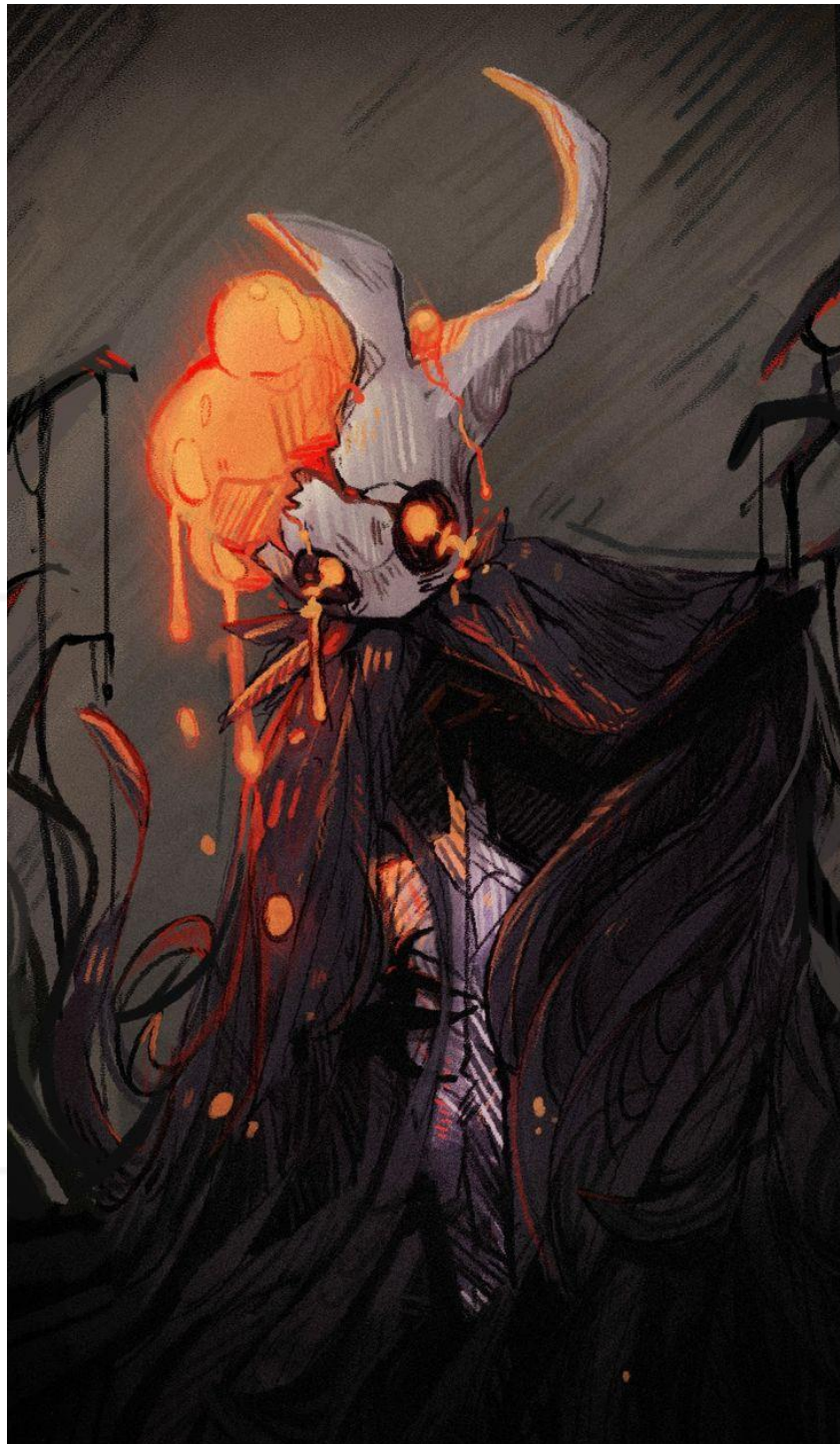
Desde la Lente del Desafío, el proyecto se evalúa en términos de equilibrio entre dificultad y justicia. El reto no debe depender de castigos arbitrarios, sino de patrones reconocibles, plataformas con lógica clara y enemigos que enseñen al jugador a leer comportamientos. Se plantea una curva de dificultad progresiva, donde cada nueva habilidad adquirida amplía tanto las posibilidades de exploración como la complejidad de los enfrentamientos. El objetivo es mantener al jugador en un estado cercano al flujo, evitando tanto la frustración excesiva como la monotonía.



La Lente de la Historia permite analizar cómo se construye la narrativa sin recurrir a exposiciones directas. En este concepto, la historia se fragmenta en descripciones de objetos, arquitectura simbólica y encuentros con personajes secundarios ambiguos. El mundo cuenta su propio pasado a través de detalles visuales y cambios en la ambientación. Esto fomenta la interpretación personal y refuerza la inmersión, ya que el jugador reconstruye activamente el significado del entorno en lugar de recibirlo de forma explícita.

Finalmente, la Lente del Espacio examina cómo se organiza el mundo jugable. El diseño propone un mapa interconectado con atajos desbloqueables, zonas verticales que refuerzan la sensación de descenso y áreas secretas que premian la observación atenta. La disposición espacial no solo cumple una función mecánica, sino también emocional: zonas amplias transmiten vulnerabilidad, pasajes estrechos generan tensión y espacios iluminados ofrecen descanso visual. Esta planificación garantiza que la arquitectura del mundo sea coherente tanto en lo funcional como en lo simbólico.

En conjunto, la aplicación de estas lentes permite fundamentar el proyecto desde múltiples dimensiones, asegurando que la experiencia no sea únicamente atractiva a nivel estético, sino también sólida en términos emocionales, mecánicos y estructurales.



Aplicación comparativa con Hollow Knight:

Hollow Knight coloca al jugador como un viajero solitario; la interfaz mínima permite que las emociones surjan desde el entorno. Para el proyecto, se replicará esa filosofía: mínima UI, sonido ambiental, lectura de mundo a través de arquitectura. **Conclusión de la lente:** El diseño debe centrarse en un jugador reflexivo y perseverante. La curva de dificultad se planificará para premiar la constancia y la capacidad de observación.

Lente del Desafío

Objetivo de la lente: evaluar si los retos son claros, significativos y estimulantes.

Preguntas clave aplicadas al proyecto:

¿Qué habilidades se pretende que el jugador desarrolle?

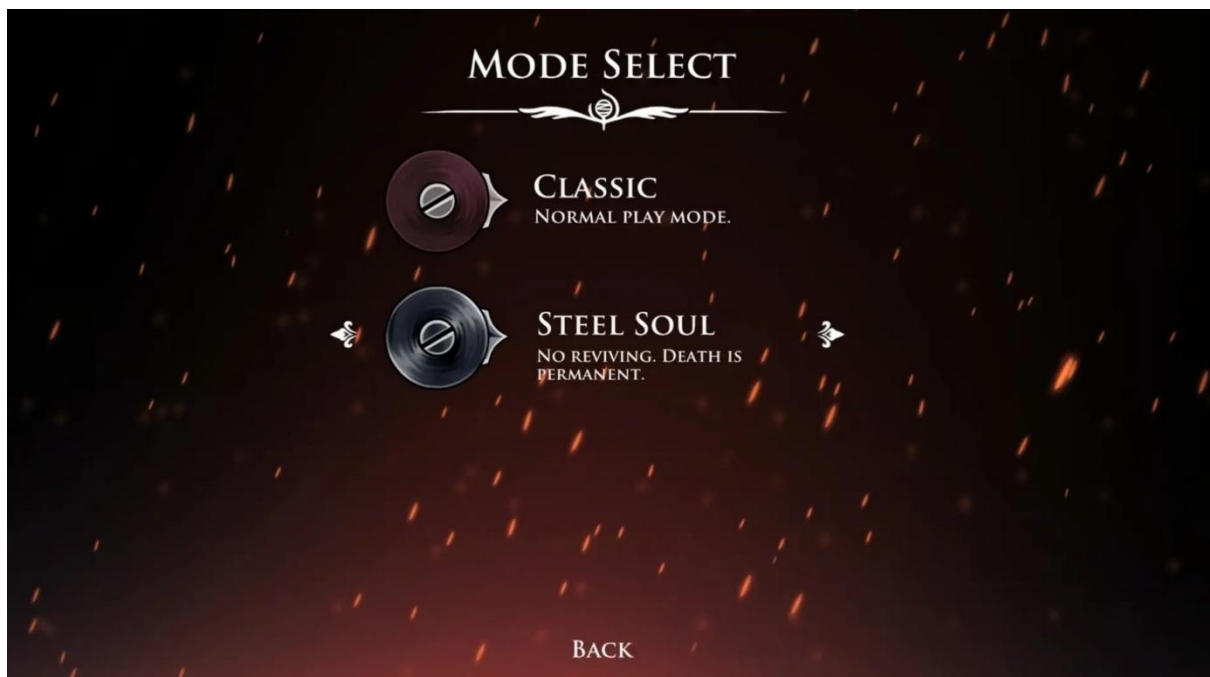
Control en plataformas precisas, gestión del riesgo, lectura del entorno, memoria espacial.

¿Los desafíos están ordenados en una progresión lógica?

Sí: se inicia con plataformas simples, luego enemigos básicos, después combinaciones de ambos, y finalmente zonas avanzadas que piden dominio completo.

¿Qué sucede cuando el jugador falla?

Se plantea un sistema de penalización moderada: pérdida parcial de un recurso que pueda recuperarse, similar a la esencia de *Hollow Knight*. Esto genera tensión y aprendizaje.



Aplicación comparativa con Hollow Knight:

Hollow Knight construye dificultad gradualmente sin tutoriales extensos. El jugador descubre y aprende mediante ensayo y error. El proyecto seguirá este principio: enseñar mediante diseño, no mediante texto.

Conclusión de la lente:

Los desafíos deben ser: comprensibles, escalonados, exigentes pero justos.

Lente de la Historia

Objetivo: asegurar que la narrativa soporte decisiones de diseño, incluso si no es explícita.

Preguntas clave:

¿Qué historia se quiere contar?

Una civilización perdida bajo tierra que dejó fragmentos de memoria repartidos en ruinas orgánicas.

¿Cómo se entregará la narrativa al jugador?

A través de:

entornos visuales

restos arquitectónicos

personajes silenciosos

objetos descriptivos

cinemáticas breves pero simbólicas

¿La historia es coherente con mecánicas y estética?

Sí: la necesidad constante de descender a capas profundas del mundo justifica el mapa vertical y los desafíos.

Comparación con Hollow Knight:

Hollow Knight evita diálogos extensos y convierte el escenario en pieza narrativa. Para el proyecto, se aplicará la misma técnica: narrativa ambiental, símbolos, ecos del pasado.

Conclusión de esta lente:

La historia debe acompañar la experiencia sin interrumpirla, permitiendo que cada jugador reconstruya su versión.



Lente del Espacio



Objetivo: analizar cómo se estructura el mapa, cómo se desplaza el jugador y cómo se guía su atención.

Preguntas clave:

¿El espacio incentiva la exploración?

Sí: el mapa se divide en regiones conectadas que requieren nuevas habilidades para desbloquear rutas alternativas.

¿El jugador identifica zonas seguras y peligrosas?

Sí: se utilizan señales visuales como cambios de color, disposición del terreno, sonidos ambientales y ritmos de iluminación.

¿Cómo se asegura claridad espacial?

Con hitos visuales, rutas principales más iluminadas y caminos secundarios más discretos.



Comparación con Hollow Knight:

Hollow Knight domina la progresión espacial con zonas reconocibles y atajos estratégicos. El proyecto replicará la estructura de zonas marcadas aunque interconectadas verticalmente.

Conclusión de la lente:

El diseño espacial debe ser progresivo, memorable, funcional y coherente con narrativa y mecánicas.

Resultados esperados

Como resultado del análisis, se ha identificado la importancia de fundamentar el diseño de videojuegos en principios teóricos sólidos mediante el uso de las lentes de Jesse Schell. Se ha logrado

estructurar una experiencia de juego coherente, donde la narrativa ambiental, el desafío progresivo y el diseño espacial convergen para generar una respuesta emocional específica en el usuario. Finalmente, se ha consolidado una planificación técnica que asegura la alineación entre las mecánicas de exploración y los objetivos psicológicos del jugador, garantizando una base sólida para el desarrollo de un prototipo funcional.



1 UNIDAD 2: Planificación de Proyectos

1.1 Practica experimental 4

Cargo:	Docente Especialista		
Nombre:	Alexis Urgilés		
Asignatura:	Diseño de Videojuegos		
Carrera:	Alexis Urgilés	Nivel:	Tercer Nivel
Estudiante:			

ACTIVIDAD PRÁCTICO EXPERIMENTAL EN EL ENTORNO ACADÉMICO

Creación de 8 modelos 3D con rigs para implementación en Unreal Engine 5

Objetivos

Modelar ocho figuras 3D básicas, aplicar rigging y prepararlas para su integración en Unreal Engine 5 como assets optimizados.

Antecedentes/Escenario

La correcta preparación de modelos incluye topología ordenada, supervisión de polígonos, jerarquías de huesos y exportación adecuada en formato compatible. Esto garantiza un desempeño óptimo en motores como Unreal.

Recursos necesarios

Software de modelado 3D.

Herramienta de rigging.

Proyecto UE5 para pruebas.

Planteamiento del problema

Se requiere crear un conjunto de modelos listos para animación e integración en un motor profesional.

Pasos por realizar

Paso 1: Crear ocho modelos con geometría limpia.

Paso 2: Generar rig para cada modelo.

Paso 3: Ajustar pesos básicos de deformación.

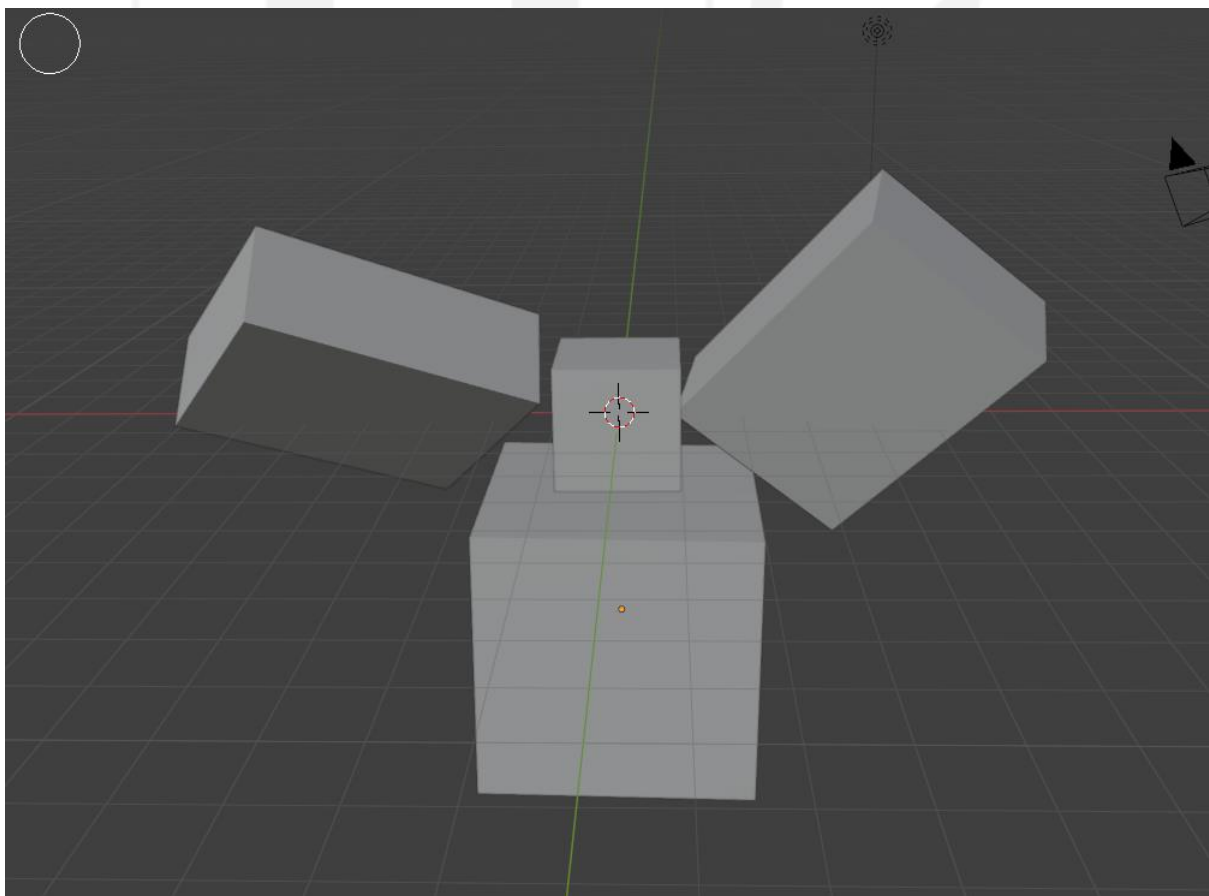
Paso 4: Exportar en formato adecuado para Unreal.

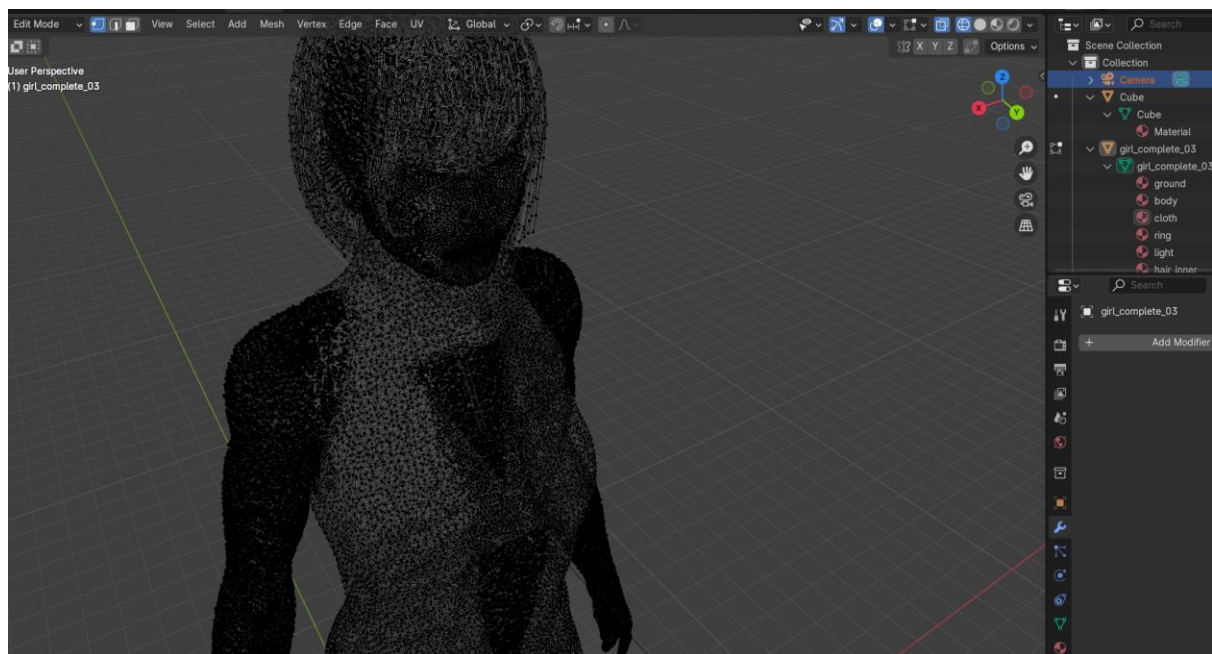
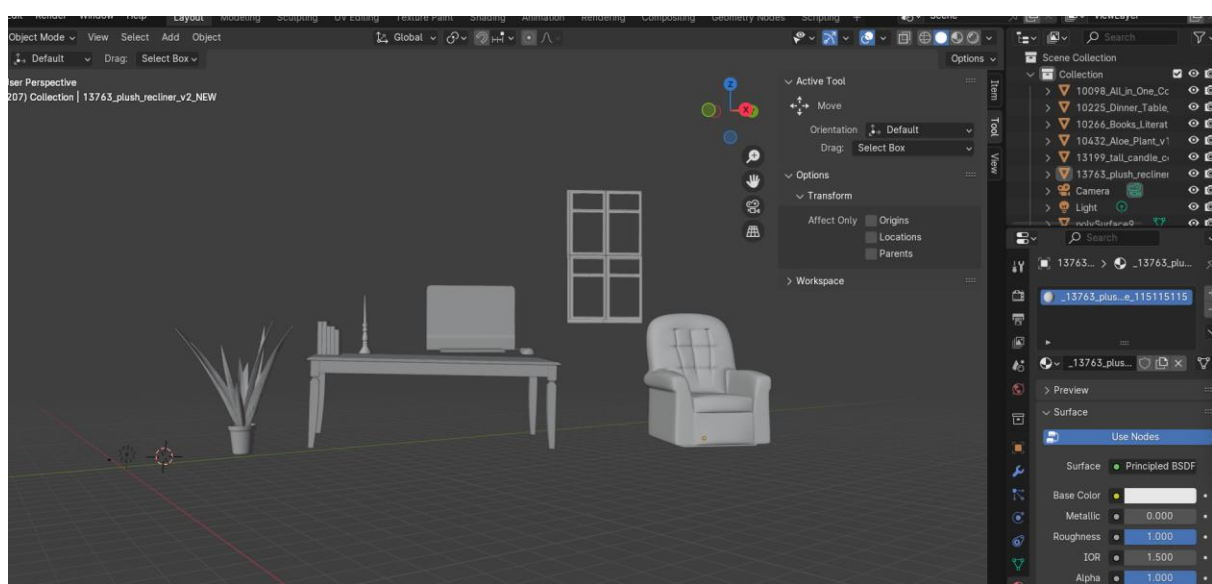
5: Importar en un proyecto de pruebas.

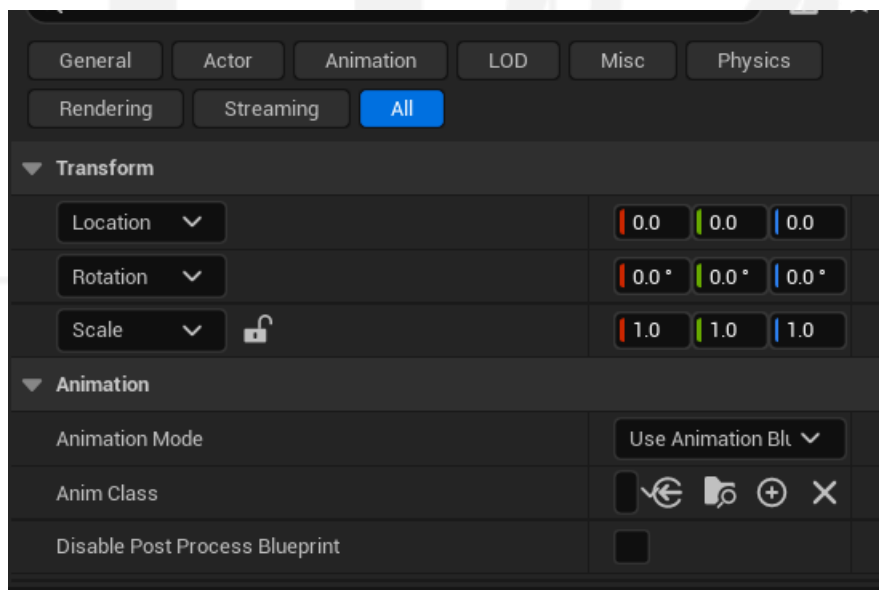
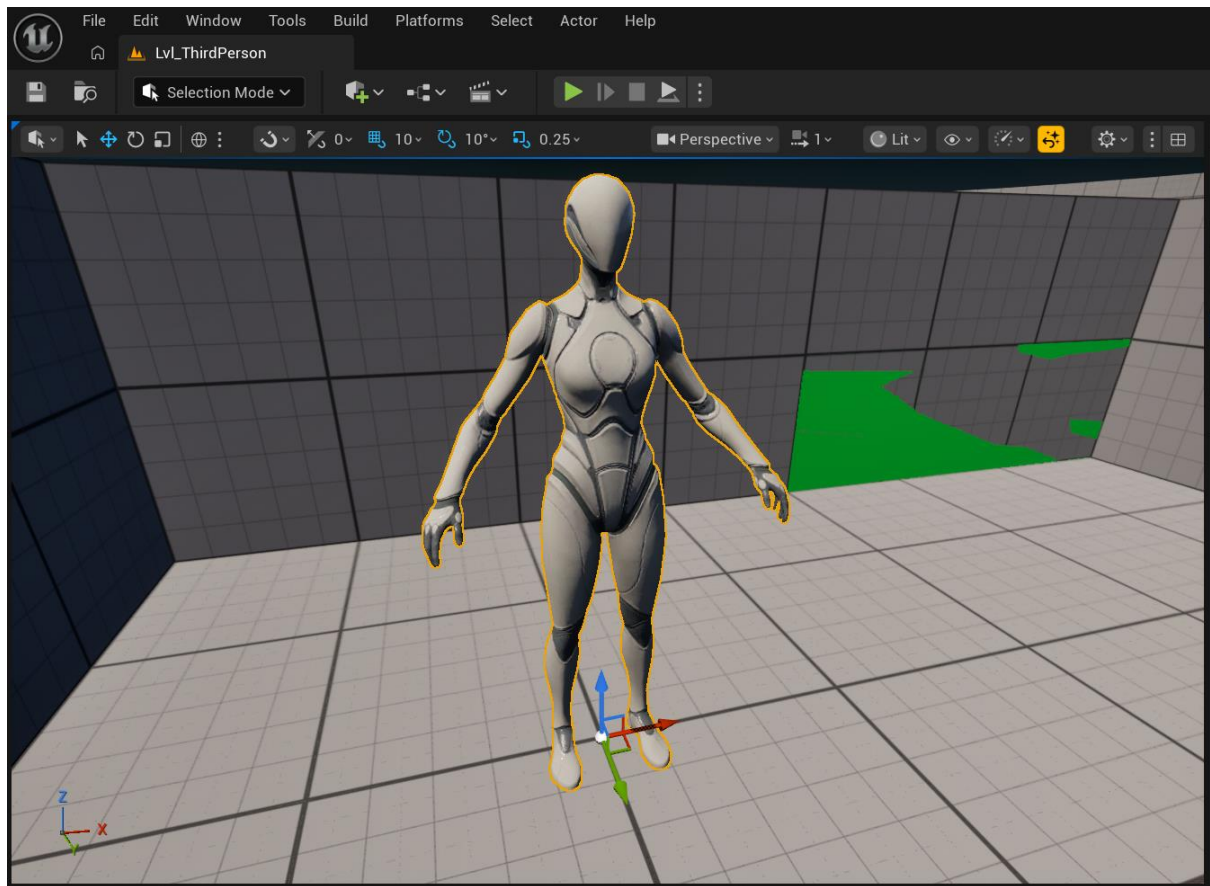
Desarrollo

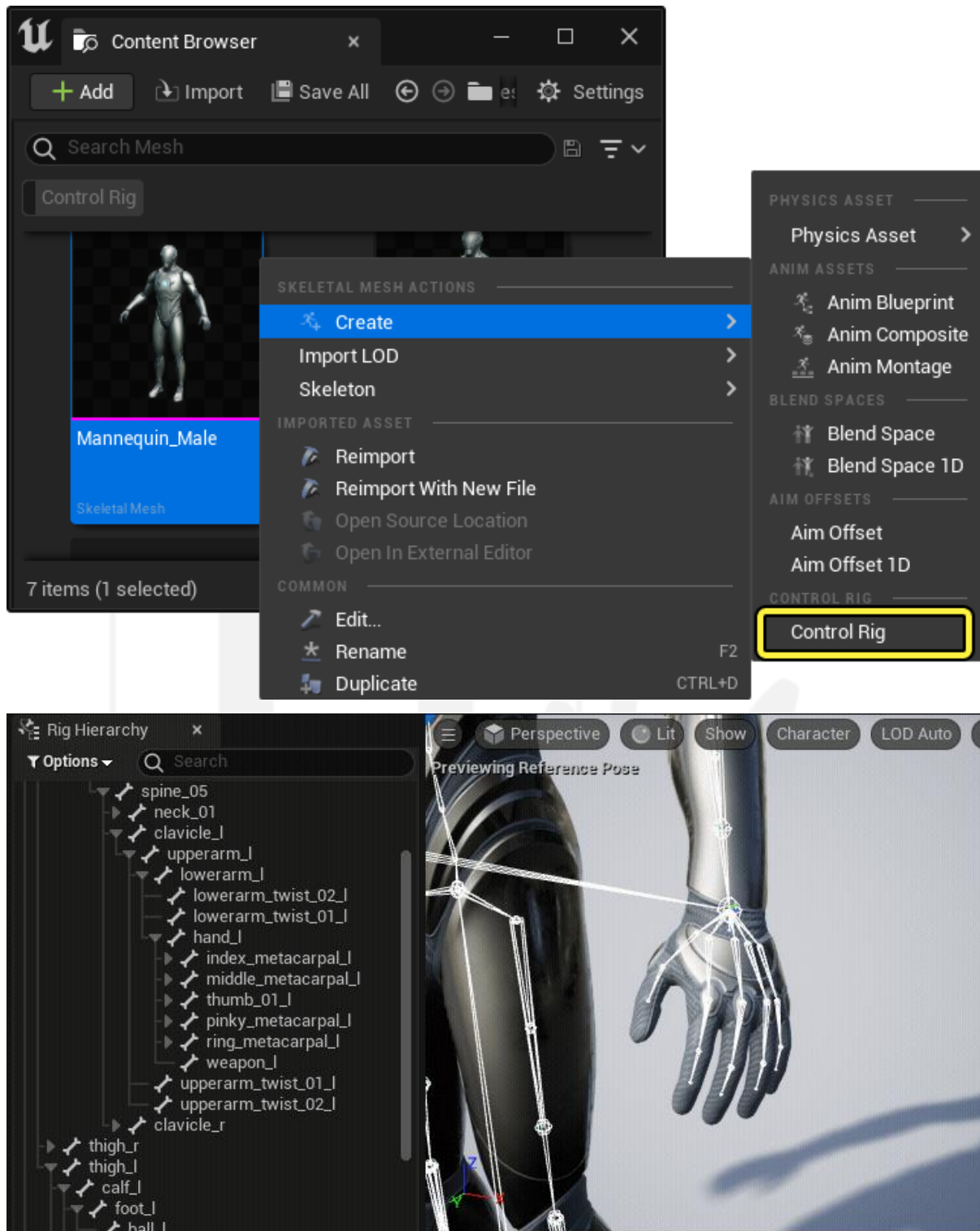
Se modelan ocho figuras con topologías simples. Se crea un esqueleto básico para cada una y se aplica la asignación de pesos para articulaciones. Se exportan los modelos y se importan dentro de Unreal para verificar su correcta lectura, escalas y orientaciones.

Nombre	Modificado...	Modificado...	File size	Compartir	Activity
florero 2	05/11/2025	ALVARADO CARDE	15 items	Shared	
florero	05/11/2025	ALVARADO CARDE	15 items	Shared	
flor 1	05/11/2025	ALVARADO CARDE	15 items	Shared	
flor 2	05/11/2025	ALVARADO CARDE	15 items	Shared	
estatua final	06/11/2025	ALVARADO CARDE	2 items	Shared	
calavera	06/11/2025	ALVARADO CARDE	15 items	Shared	









Resultados esperados

Se espera que el estudiante logre identificar la importancia de la topología limpia y la jerarquía de huesos en la preparación de activos para motores de alto rendimiento. Se ha logrado la creación de ocho modelos 3D con rigging funcional y pesos de deformación ajustados, garantizando una transición fluida hacia Unreal Engine 5. Finalmente, se ha validado la interoperabilidad de

formatos, asegurando que las escalas, orientaciones y mallas esqueléticas cumplan con los estándares técnicos exigidos para la animación y optimización en tiempo real.



1.2 Practica experimental 5

Cargo:	Docente Especialista		
Nombre:	Alexis Urgilés		
Asignatura:	Diseño de Videojuegos		
Carrera:	Alexis Urgilés	Nivel:	Tercer Nivel
Estudiante:			

ACTIVIDAD PRÁCTICO EXPERIMENTAL EN EL ENTORNO ACADÉMICO

Creación de un mundo en Unreal Engine con una cueva utilizando un solo asset transformado

Objetivos

Crear un entorno tipo cueva aplicando transformaciones diversas a un solo asset para simular variedad espacial.

Antecedentes/Escenario

Los motores permiten reducir carga visual reutilizando un mismo asset mediante escalado, rotación, duplicación y deformación leve, lo cual optimiza tiempo y recursos sin sacrificar calidad.

Recursos necesarios

Un solo asset de roca o estructura natural.

Proyecto Unreal Engine.

Planteamiento del problema

Se debe construir un entorno que dé sensación de profundidad utilizando transformaciones creativas del mismo asset.

Pasos por realizar

Paso 1: Importar el asset.

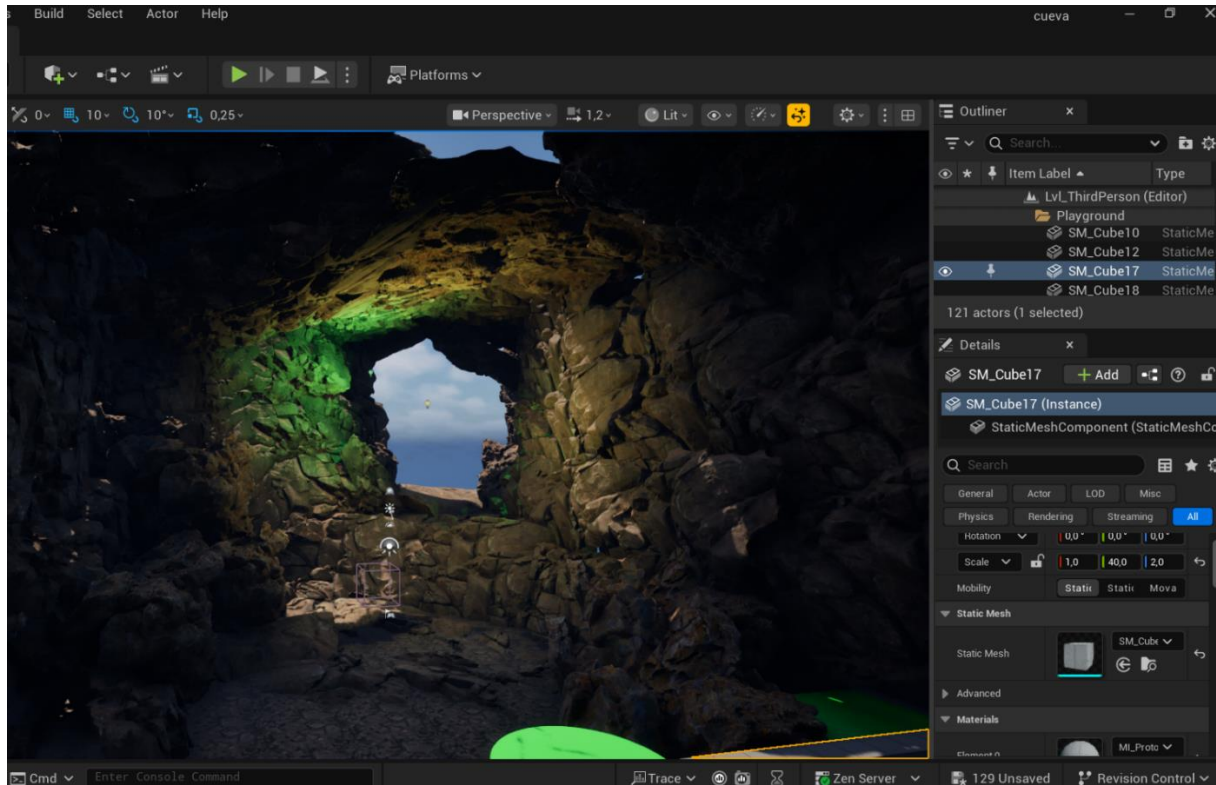
Paso 2: Colocarlo en diferentes posiciones generando paredes y techo.

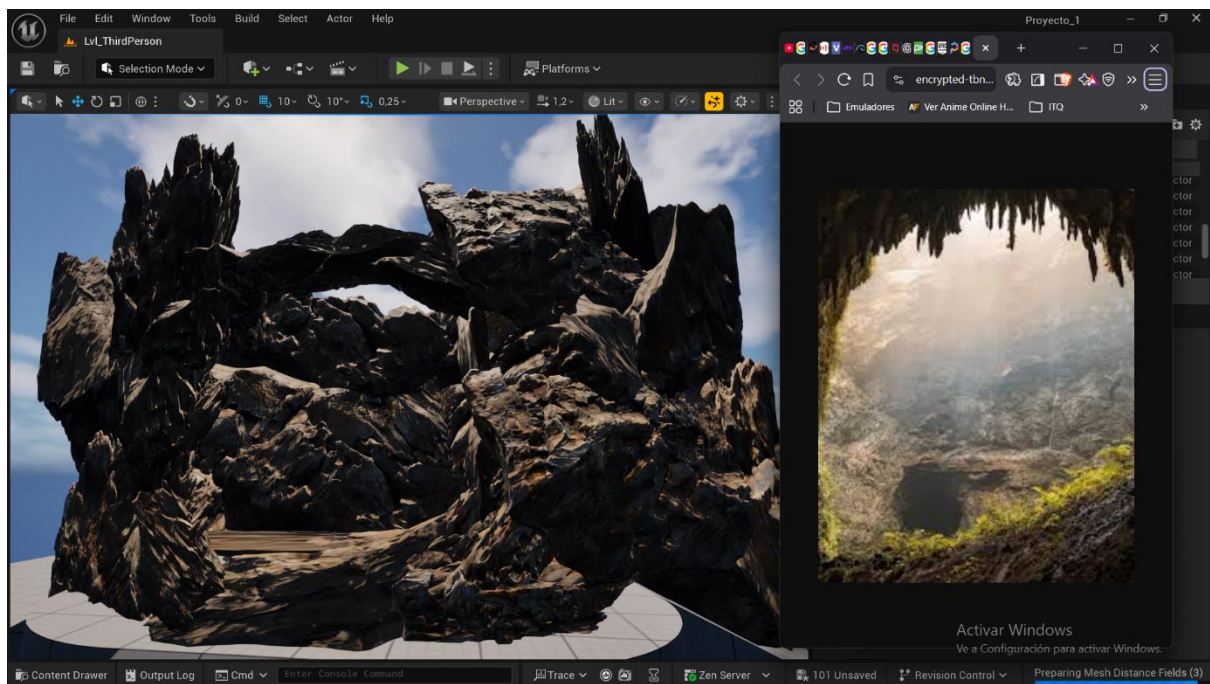
Paso 3: Aplicar variaciones de escala y rotación.

Paso 4: Ajustar iluminación básica.

Desarrollo

Se importa un asset de roca. Se colocan múltiples copias modificando escala y rotación para crear pasillos, entradas y espacios amplios. Se ajusta iluminación puntual y ambiental para simular profundidad y dirección de luz. Se revisa la composición general para asegurar coherencia en el entorno.





Resultados Esperados

Al finalizar la actividad, el estudiante será capaz de construir un entorno tipo cueva utilizando un solo asset modificado estratégicamente mediante escalas, rotaciones y duplicaciones. Se espera que demuestre comprensión del uso eficiente de recursos, composición espacial, lectura volumétrica y coherencia ambiental dentro del motor. El mundo creado deberá transmitir sensación de profundidad y continuidad, integrar iluminación básica funcional y reflejar criterios fundamentales de diseño de niveles, optimización visual y creatividad en la reutilización de elementos.

1.3 Practica experimental 6

Cargo:	Docente Especialista		
Nombre:	Alexis Urgilés		
Asignatura:	Diseño de Videojuegos		
Carrera:	Alexis Urgilés	Nivel:	Tercer Nivel
Estudiante:			

ACTIVIDAD PRÁCTICO EXPERIMENTAL EN EL ENTORNO ACADÉMICO

Composición con elementos básicos y luces en Unreal Engine

Objetivos

Crear una composición visual aplicando figuras básicas y luces para comprender cómo el motor construye atmósferas.

Antecedentes/Escenario

Las composiciones iniciales permiten entender valores fundamentales como contraste, color, dirección de luz y jerarquía visual.

Recursos necesarios

Unreal Engine.

Elementos básicos como cubos, esferas o cilindros.

Planteamiento del problema

Se requiere realizar una composición equilibrada mediante formas simples y distintos tipos de luces.

Pasos por realizar

Paso 1: Colocar figuras básicas.

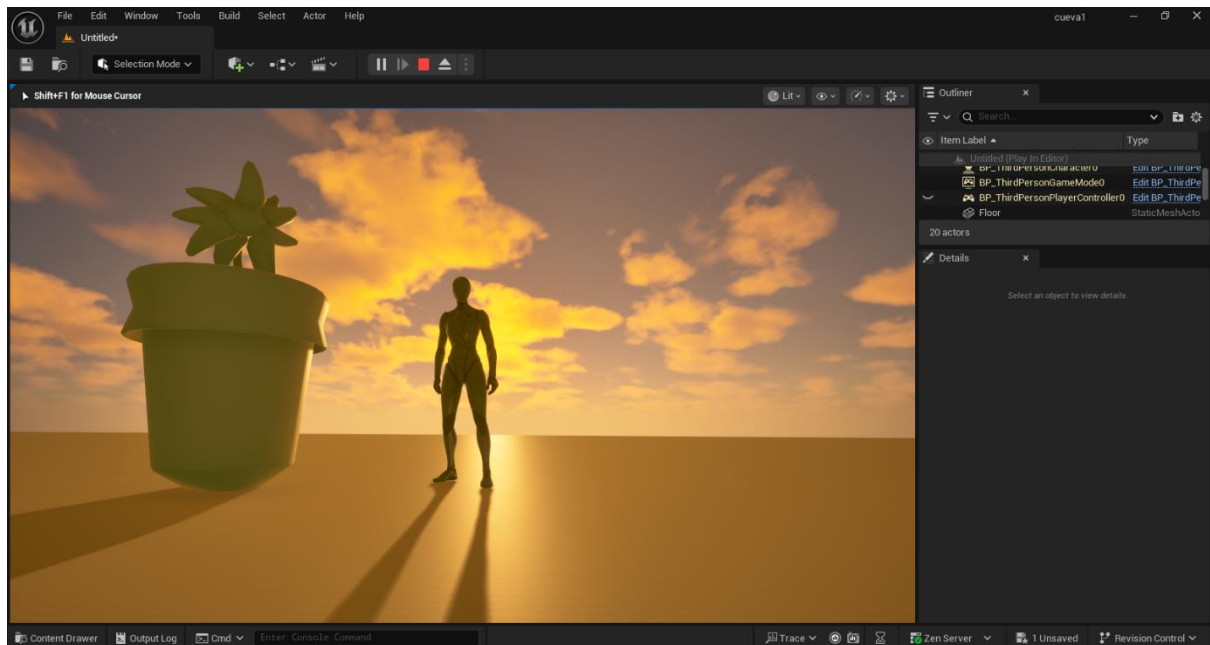
Paso 2: Ajustar posiciones y alturas.

Paso 3: Añadir luces principales.

Paso 4: Probar variaciones de intensidad y color.

Desarrollo

Se distribuyen figuras básicas en la escena buscando equilibrio visual. Se añaden luces direccionales, puntuales o de área según convenga. Se ajustan colores y valores de exposición para conseguir un ambiente coherente. Se capturan diferentes versiones para comparar resultados.



Resultados esperados

Al completar la actividad, el estudiante será capaz de construir una composición visual coherente en Unreal Engine utilizando únicamente figuras básicas y configuraciones esenciales de

iluminación. Se espera que demuestre comprensión de principios fundamentales como equilibrio, contraste, color, jerarquía y atmósfera. El estudiante deberá manipular diferentes tipos de luces, ajustar intensidades y direcciones, y analizar cómo estos elementos influyen en la percepción espacial. La escena final debe evidenciar criterio compositivo y un uso intencional de los recursos disponibles.



2 UNIDAD 3: Rigging

2.1 Practica experimental 7

Cargo:	Docente Especialista		
Nombre:	Alexis Urgilés		
Asignatura:	Diseño de Videojuegos		
Carrera:	Alexis Urgilés	Nivel:	Tercer Nivel
Estudiante:			

ACTIVIDAD PRÁCTICO EXPERIMENTAL EN EL ENTORNO ACADÉMICO

Investigación sobre el Event Graph en UE5.6 y sus funciones principales

Objetivos

Investigar el funcionamiento del Event Graph, sus nodos principales y su papel en la lógica de Blueprints.

Antecedentes/Escenario

El Event Graph centraliza la ejecución de eventos y funciones dentro de Blueprints, permitiendo definir comportamiento sin programación textual.

Recursos necesarios

Proyecto UE5.6.

Documentación sobre Blueprints.

Planteamiento del problema

Se requiere documentar el uso del Event Graph y comprender su relación con eventos del motor y llamadas a funciones.

Pasos por realizar

Paso 1: Abrir un Blueprint.

Paso 2: Identificar nodos principales.

Paso 3: Investigar usos comunes.

Paso 4: Documentar conclusiones.

Desarrollo

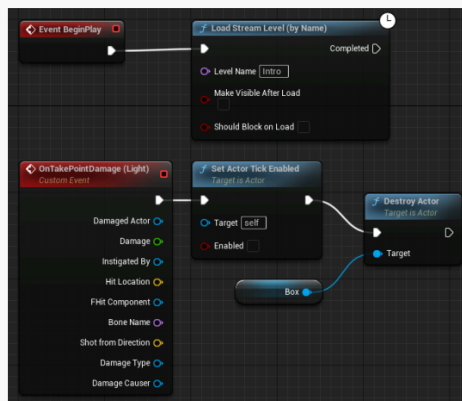
Se abre un Blueprint y se identifica el Event Graph como espacio donde se manejan eventos como BeginPlay, Tick o OnOverlap. Se investiga cómo estos nodos inician flujos lógicos y cómo pueden conectarse con variables, funciones o macros. Se documentan ejemplos frecuentes como controles de jugador, animaciones o interacción.

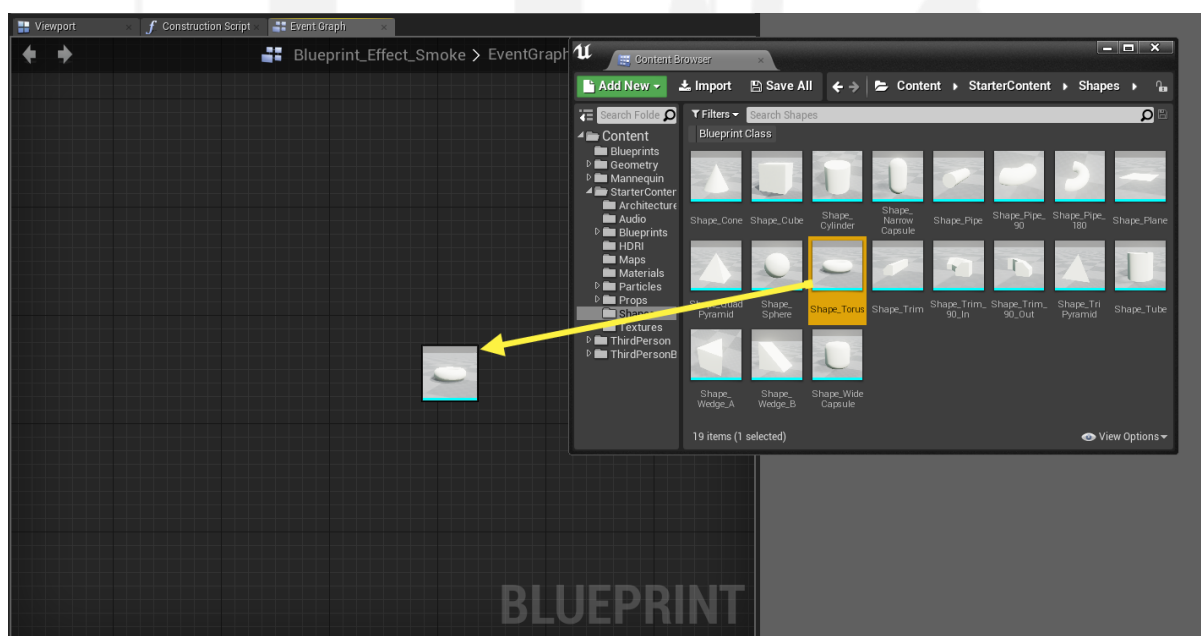
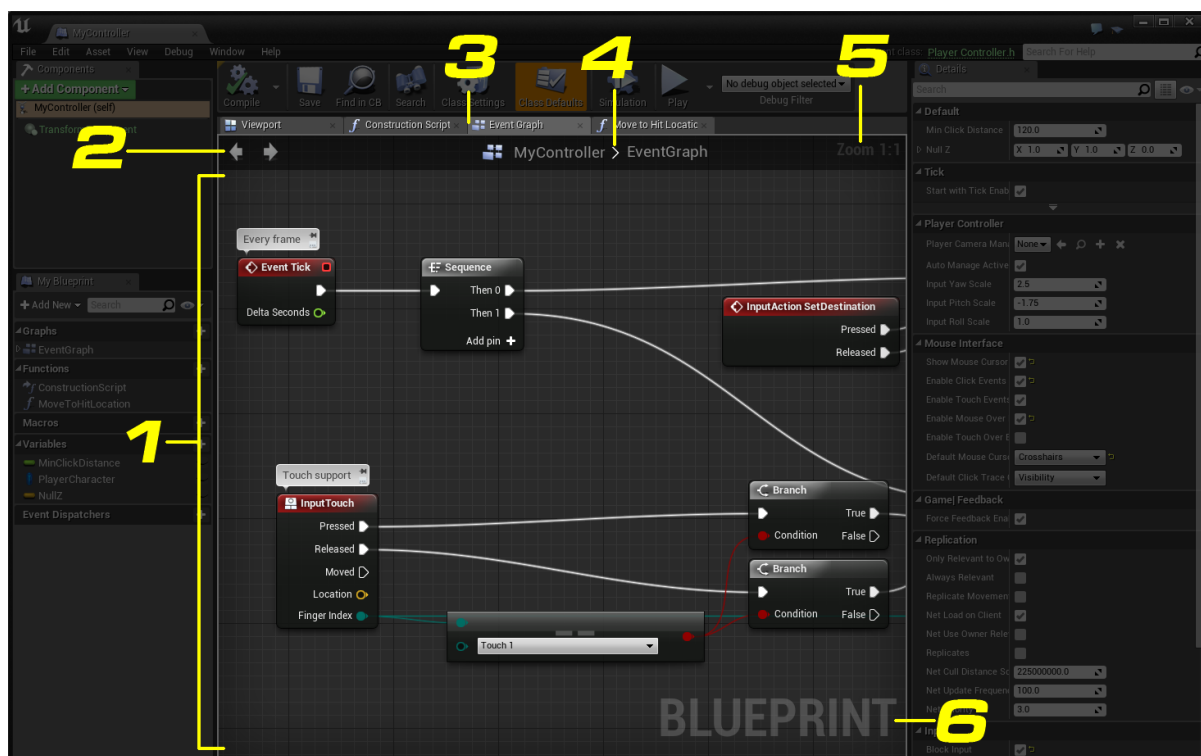
EVENT GRAPH

Es un nodo grafico que contiene nodos que utiliza eventos y funciones que realizan acciones en respuesta de un evento asociado al Blueprint.

El event graphic de un blueprint de nivel contiene un nodo grafico, los eventos y funciones que daran respuesta a eventos de juego por ejemplo interacciones de actores con el mundo.

EVENT GRAPH





Resultados esperados

Al finalizar la actividad, el estudiante será capaz de comprender el funcionamiento esencial del Event Graph en Unreal Engine 5.6, identificando sus nodos principales y su rol dentro de la lógica basada en Blueprints. Se espera que distinga cómo eventos como BeginPlay, Tick y las colisiones activan diferentes flujos lógicos, y que analice cómo estos se integran con variables, funciones y

macros. El estudiante demostrará capacidad para explicar el propósito del Event Graph y su importancia en la construcción del comportamiento interactivo dentro del motor.



2.2 Practica experimental 8

Cargo:	Docente Especialista		
Nombre:	Alexis Urgilés		
Asignatura:	Diseño de Videojuegos		
Carrera:	Alexis Urgilés	Nivel:	Tercer Nivel
Estudiante:			

ACTIVIDAD PRÁCTICO EXPERIMENTAL EN EL ENTORNO ACADÉMICO

Configuración de luces principales en Unreal Engine: Directional Light, Sky Light y Post Process

Volume

Objetivos

Configurar las luces principales del motor para comprender cómo afectan la iluminación global y local.

Antecedentes/Escenario

La iluminación determina el ambiente del mundo virtual. La luz direccional simula el sol, la luz de cielo captura iluminación ambiental y el Post Process controla efectos visuales globales.

Recursos necesarios

Proyecto Unreal Engine.

Planteamiento del problema

Se requiere ajustar correctamente las luces principales para obtener una escena equilibrada.

Pasos por realizar

Paso 1: Colocar una Directional Light.

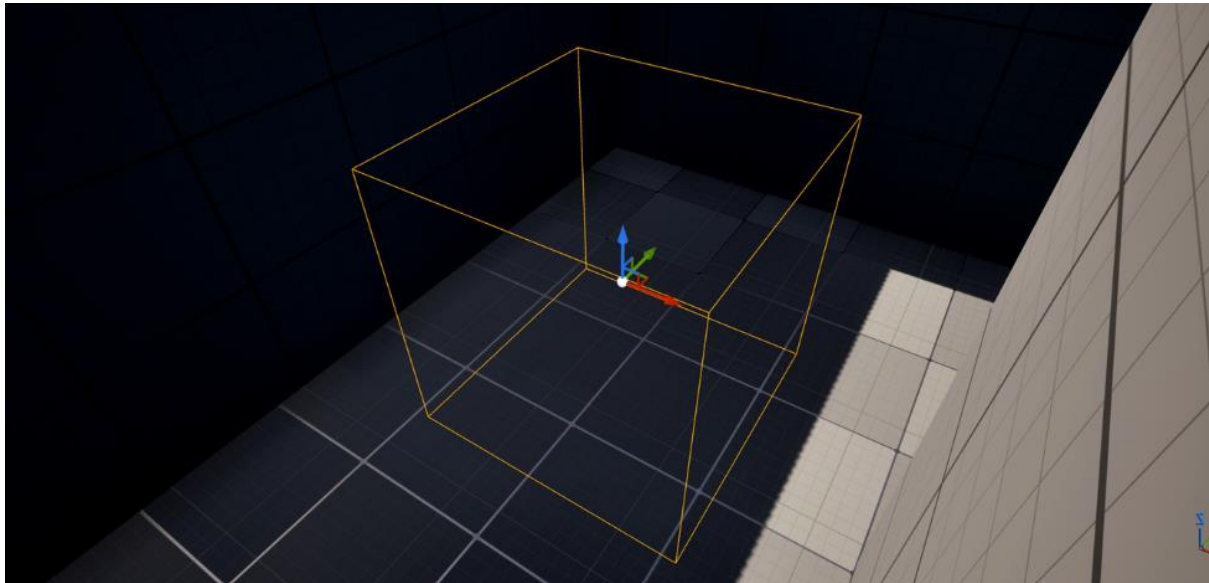
Paso 2: Configurar Sky Light.

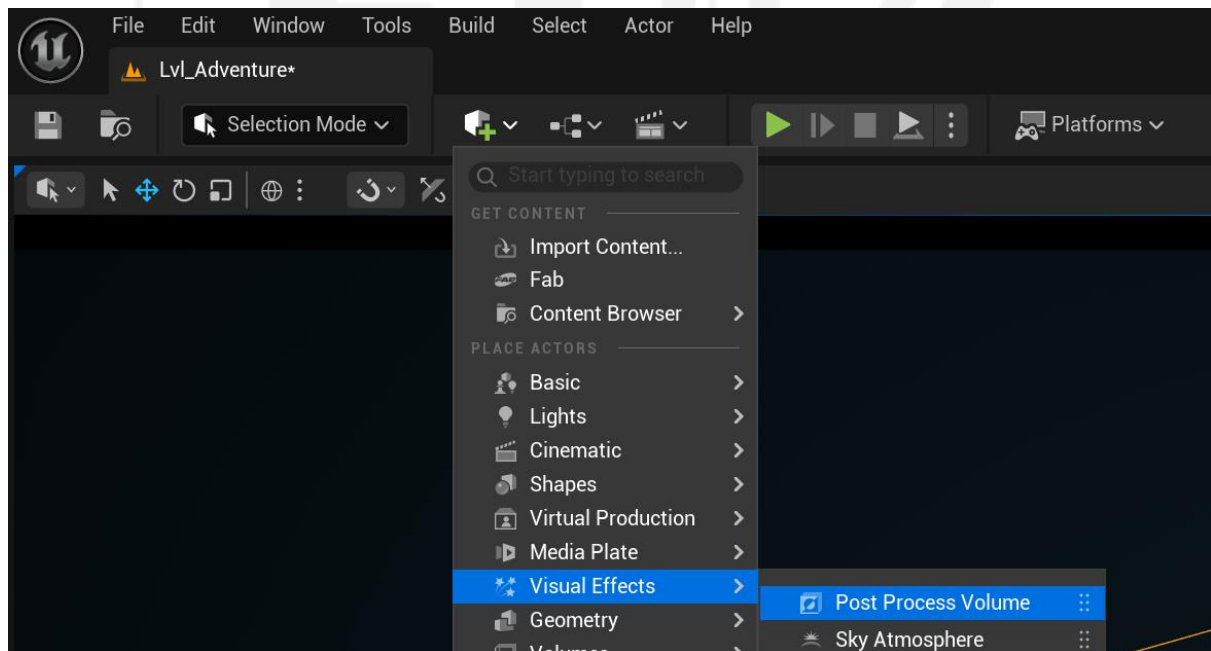
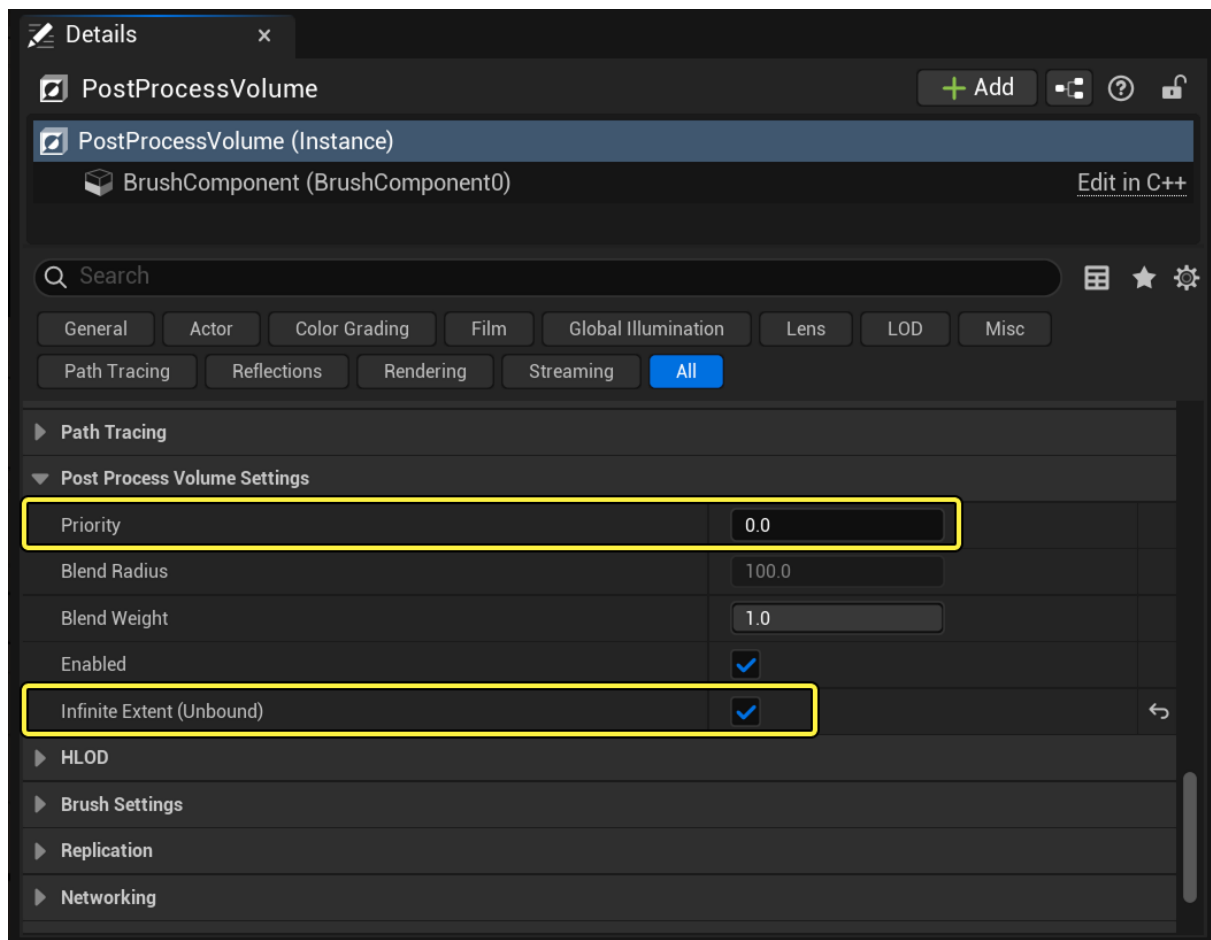
Paso 3: Añadir Post Process Volume.

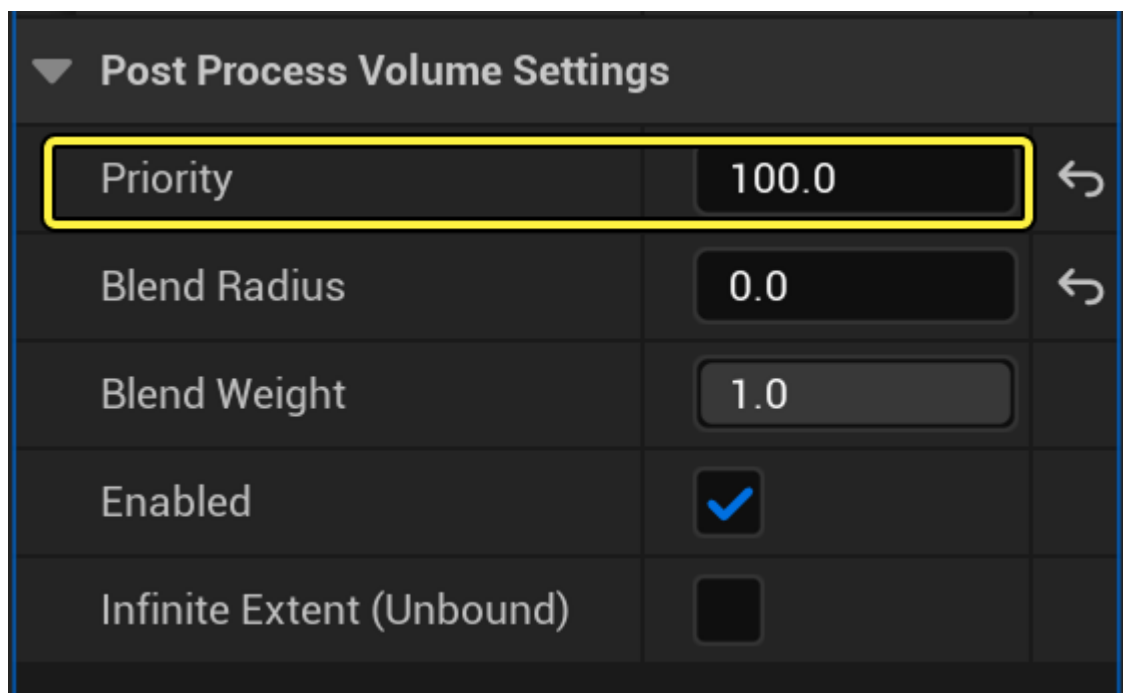
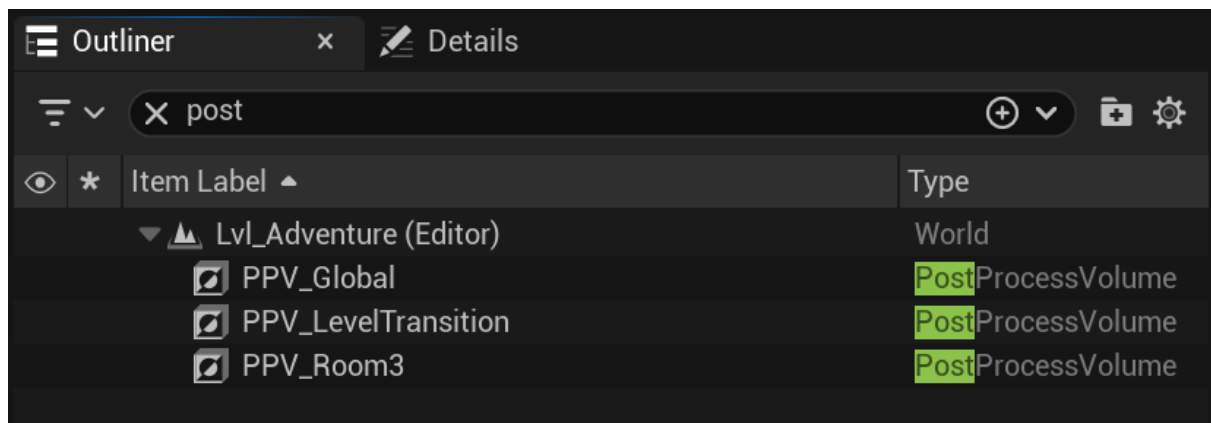
Paso 4: Ajustar exposición, color y contrastes.

Desarrollo

Se añade una luz direccional y se ajusta su intensidad y ángulo. Se coloca una Sky Light para añadir iluminación ambiental suave. Se agrega un Post Process Volume para controlar exposición, bloom, saturación y temperatura. Se experimenta con diferentes configuraciones para obtener la atmósfera deseada.









Resultados esperados

Al concluir la actividad, el estudiante será capaz de configurar de manera correcta y funcional los tres componentes esenciales de iluminación en Unreal Engine: Directional Light, Sky



Light y Post Process Volume. Se espera que comprenda cómo cada uno influye en la percepción visual del entorno, ajustando intensidad, color, exposición y contraste para obtener una escena coherente y equilibrada. El estudiante demostrará dominio básico en la creación de atmósferas, identificando cómo la interacción entre estas luces define profundidad, claridad y estilo visual dentro del mundo virtual.



2.3 Practica experimental 9

Cargo:	Docente Especialista		
Nombre:	Alexis Urgilés		
Asignatura:	Diseño de Videojuegos		
Carrera:	Alexis Urgilés	Nivel:	Tercer Nivel
Estudiante:			

ACTIVIDAD PRÁCTICO EXPERIMENTAL EN EL ENTORNO ACADÉMICO

Implementación de mapas de textura para un material con alphas y máscaras

Objetivos

Los materiales modernos utilizan múltiples texturas para definir color, normal maps, roughness y elementos basados en alphas para transparencias y mezcla.

Antecedentes/Escenario

Un motor de videojuegos integra múltiples subsistemas: renderizado, física, audio, animación, entrada, scripting y administración de activos.

Recursos necesarios

Texturas con alpha.

Editor de materiales en Unreal Engine.

Planteamiento del problema

Implementar un material funcional que combine texturas y máscaras.

Pasos por realizar

Paso 1: Importar texturas.

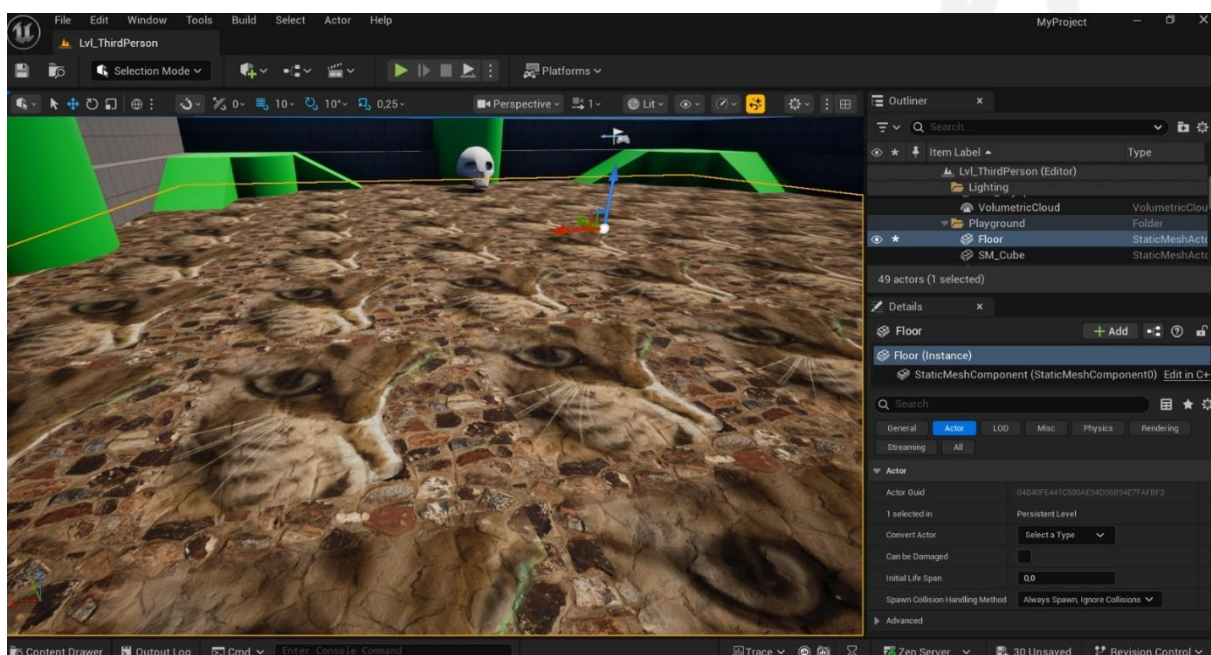
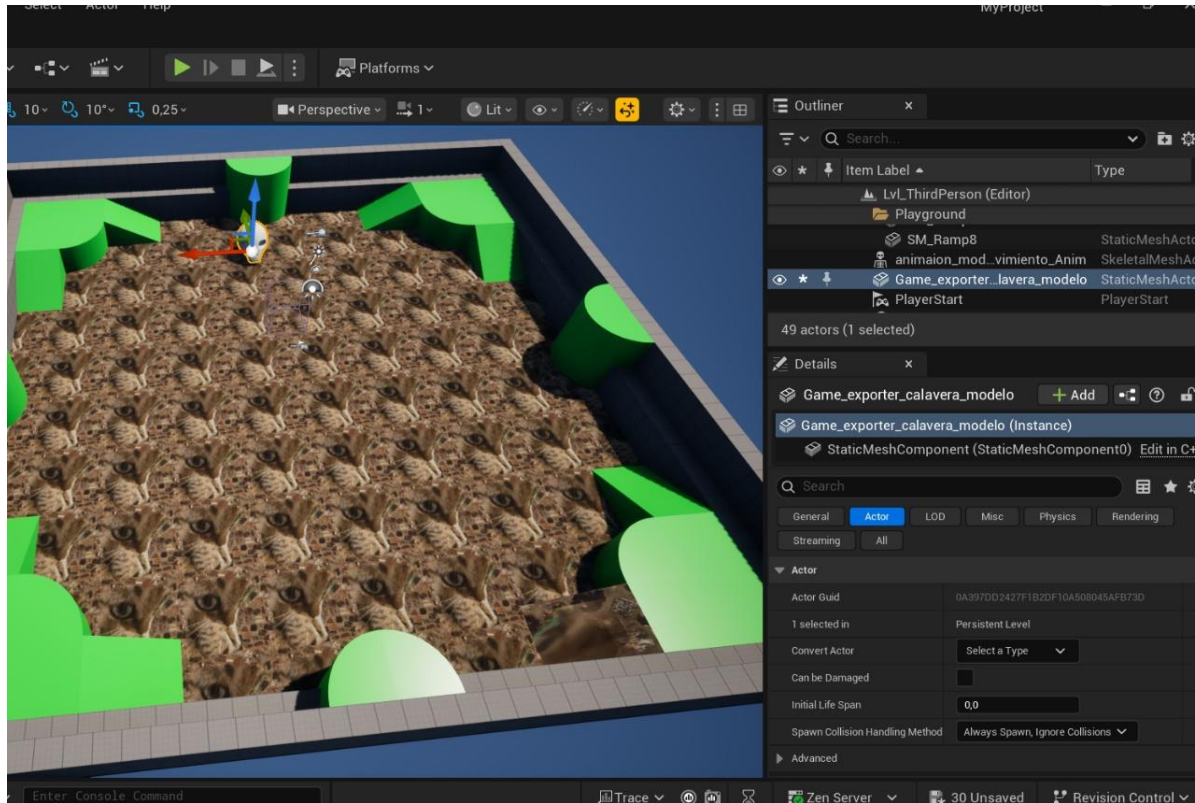
Paso 2: Crear material nuevo.

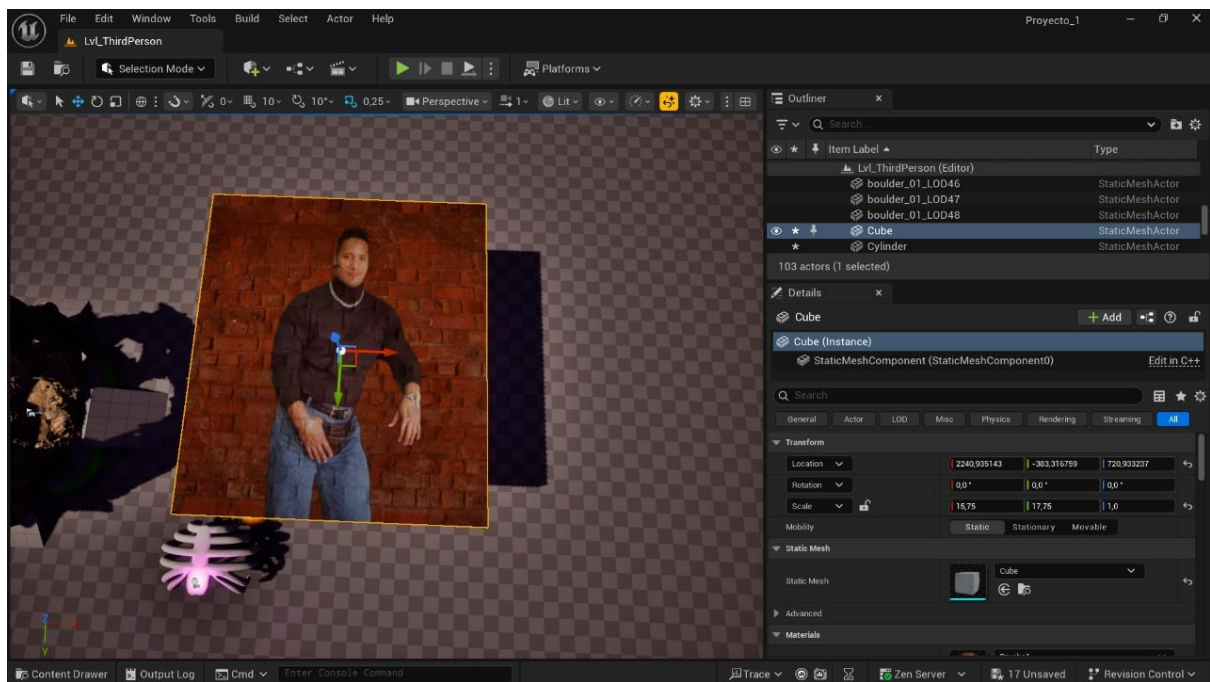
Paso 3: Conectar mapas de color, normal y roughness.

Paso 4: Añadir máscaras mediante alpha.

Desarrollo

Se importan todas las texturas necesarias. Se crea un material y se conectan mapas a los nodos correspondientes. Se utiliza el canal alpha para crear zonas transparentes o de mezcla. Se prueba el material aplicándolo a diferentes objetos para evaluar cómo reacciona a la iluminación.





Resultados esperados

Al finalizar la actividad, el estudiante será capaz de crear un material completo en Unreal Engine utilizando mapas de textura avanzados, integrando correctamente color, normal maps, roughness y canales alpha. Se espera que comprenda cómo cada textura modifica las propiedades visuales del material y cómo las máscaras permiten generar transparencias o mezclas controladas. Además, podrá evaluar el comportamiento del material frente a diferentes fuentes de luz, demostrando dominio básico en la construcción de superficies realistas y funcionales para su uso en entornos interactivos.

3 UNIDAD 4: Programación en Unreal Engine

3.1 Practica experimental 10

Cargo:	Docente Especialista		
Nombre:	Alexis Urgilés		
Asignatura:	Diseño de Videojuegos		
Carrera:	Alexis Urgilés	Nivel:	Tercer Nivel
Estudiante:			

ACTIVIDAD PRÁCTICO EXPERIMENTAL EN EL ENTORNO ACADÉMICO

Creación e implementación de widgets para la interfaz de usuario

Objetivos

Diseñar e implementar un widget funcional como parte de la interfaz del videojuego.

Antecedentes/Escenario

a interfaz permite comunicar información clave al jugador. Unreal ofrece herramientas visuales para crear HUDs, indicadores y menús.

Recursos necesarios

Editor UMG en Unreal Engine.

Planteamiento del problema

Crear un widget funcional y vincularlo con el jugador.

Pasos por realizar

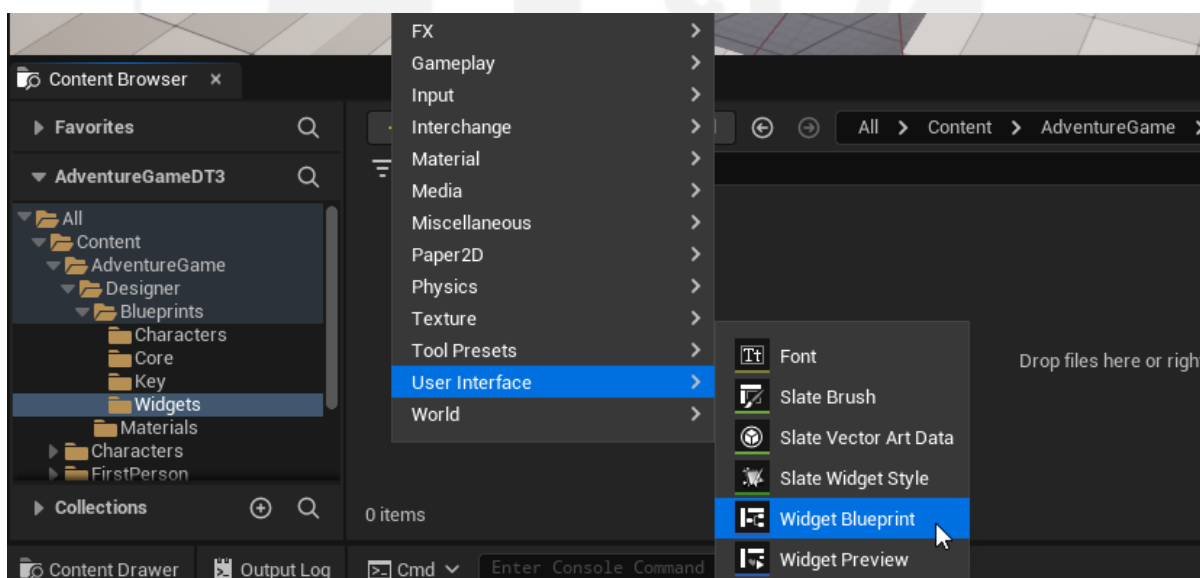
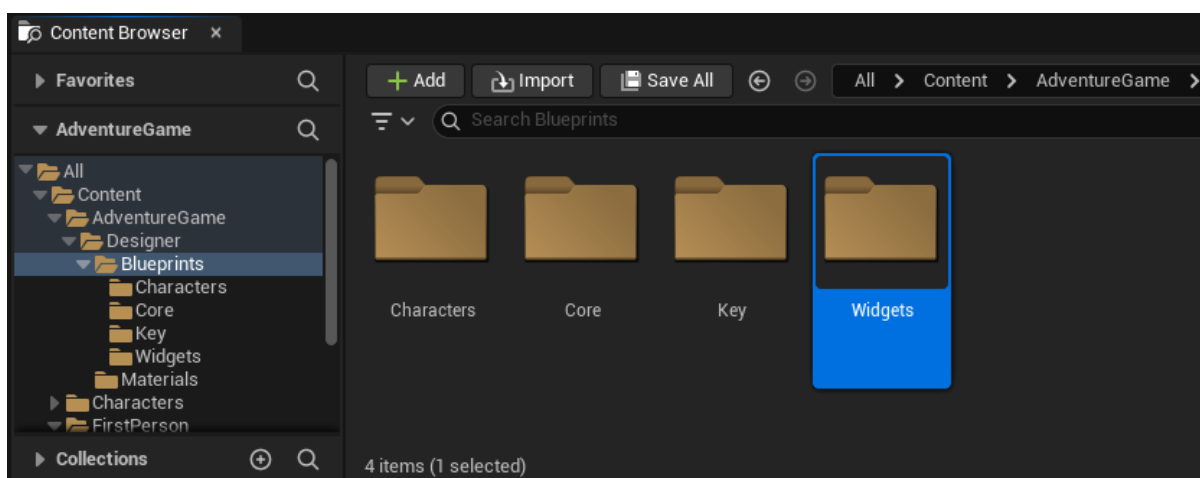
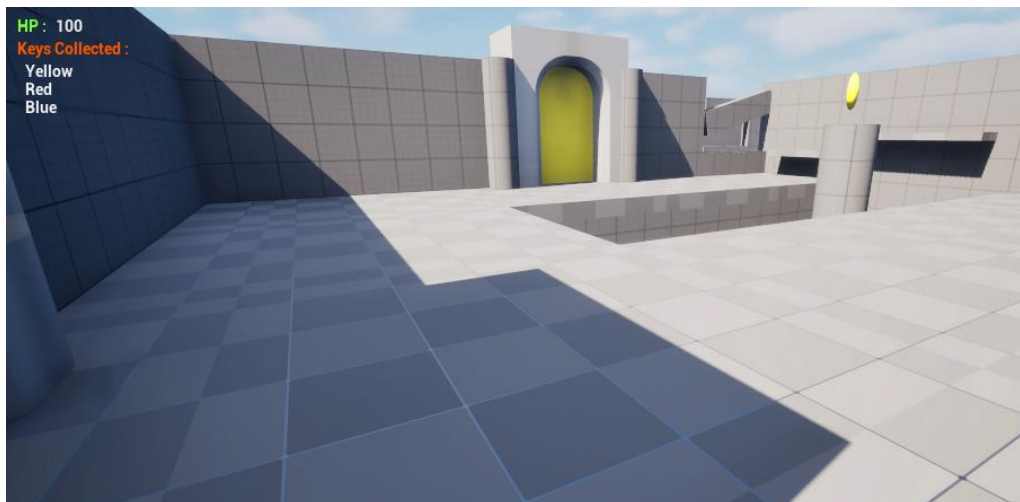
Paso 1: Crear widget.

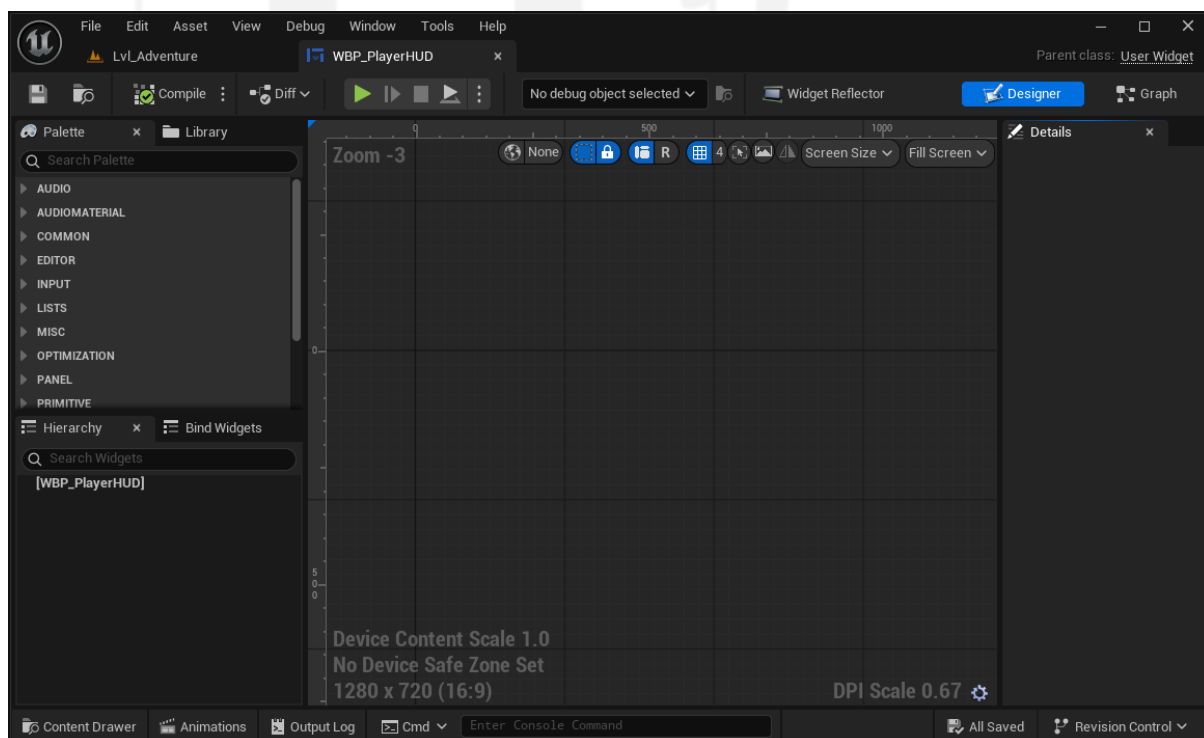
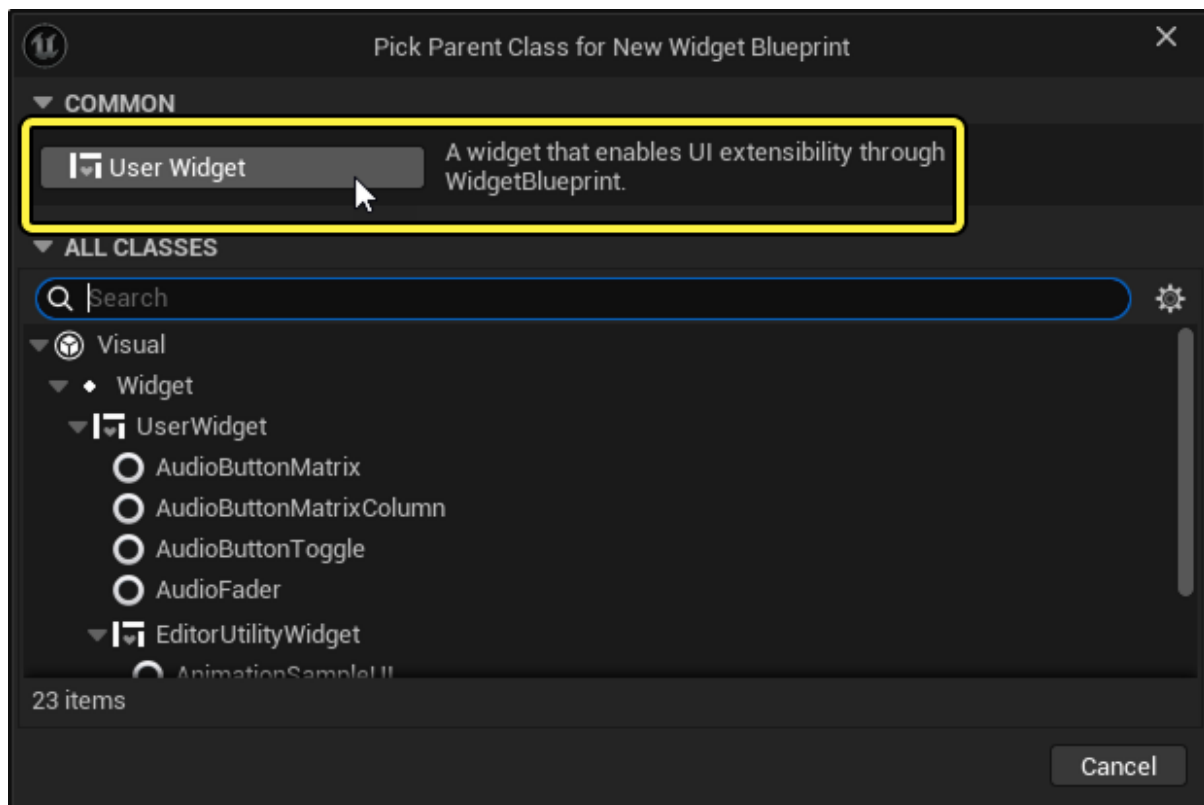
Paso 2: Añadir elementos visuales.

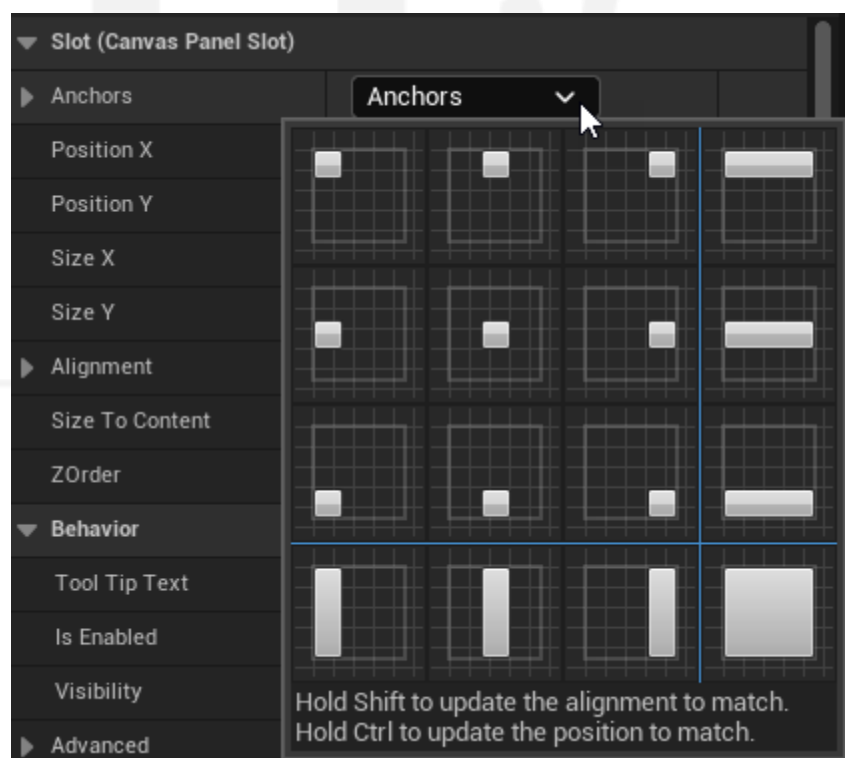
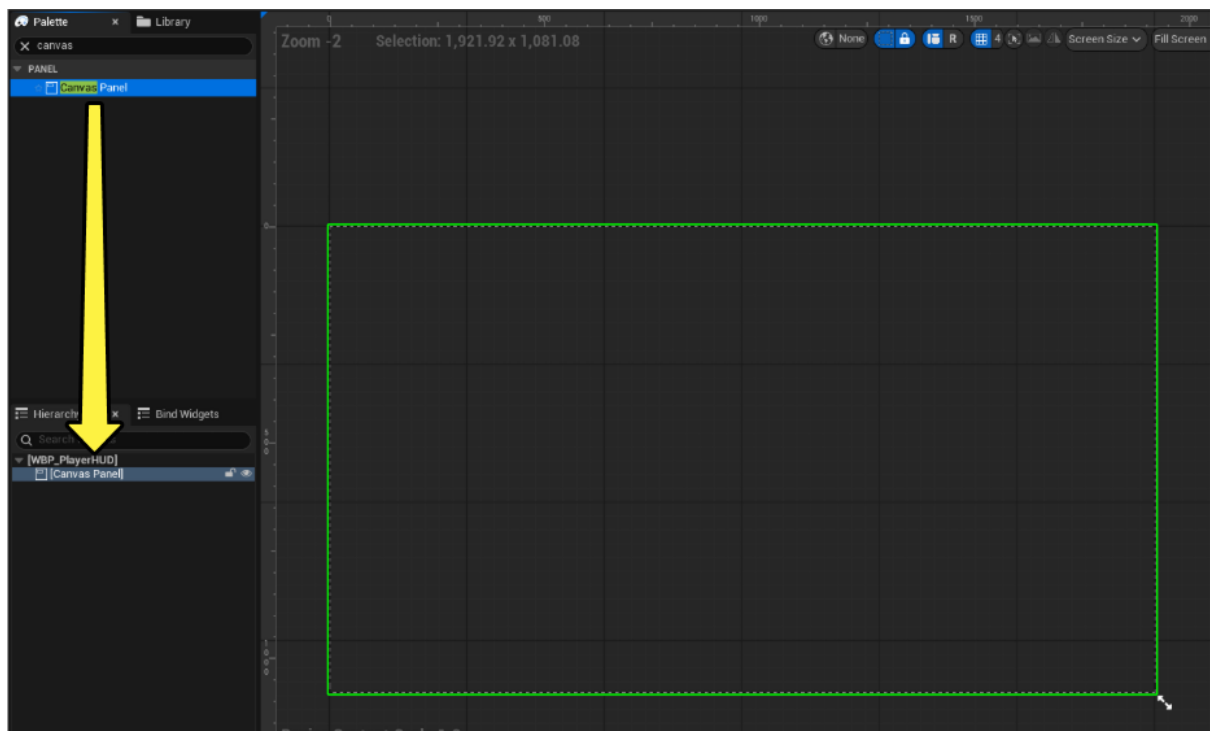
Paso 3: Vincularlo con el HUD.

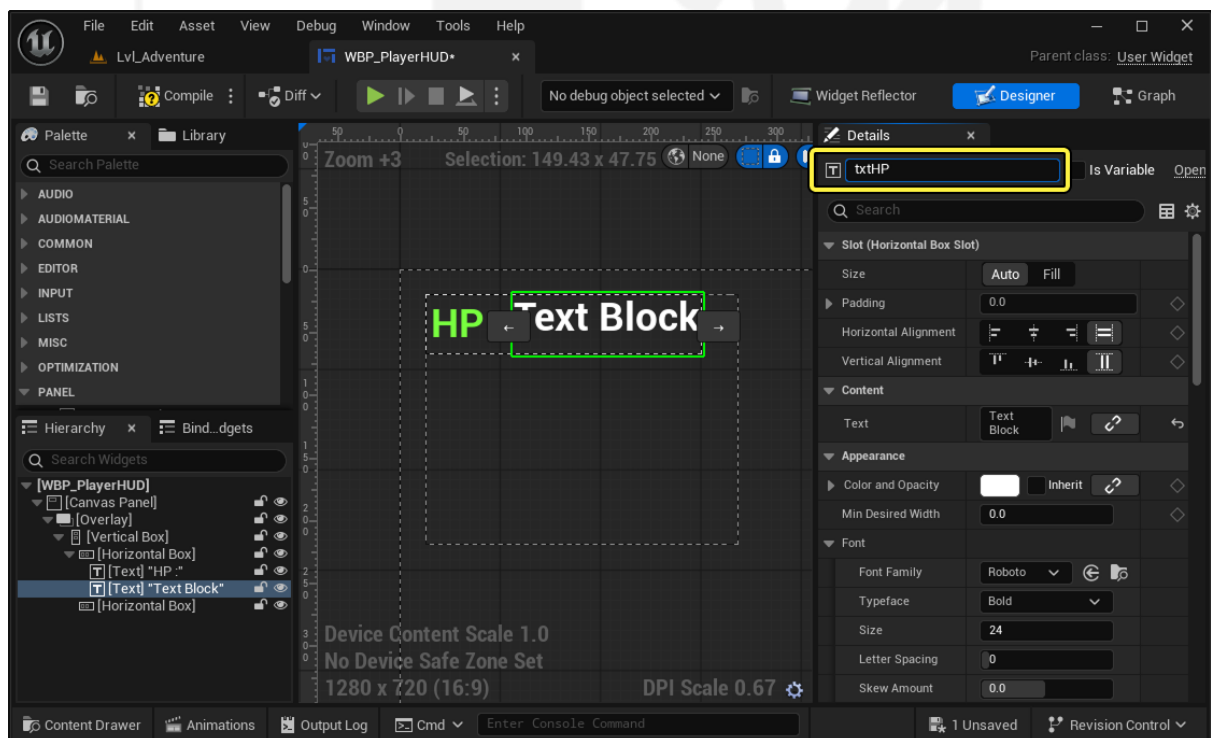
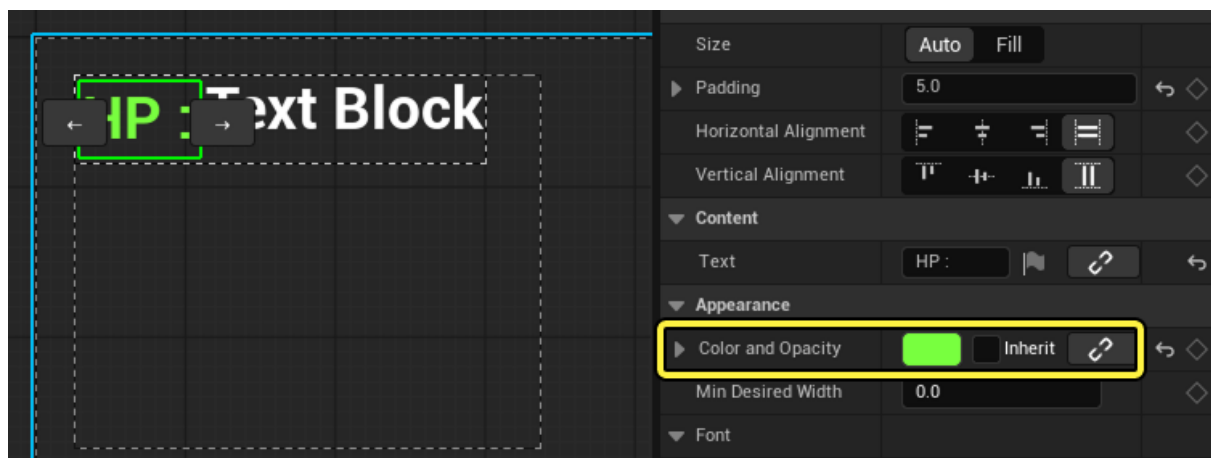
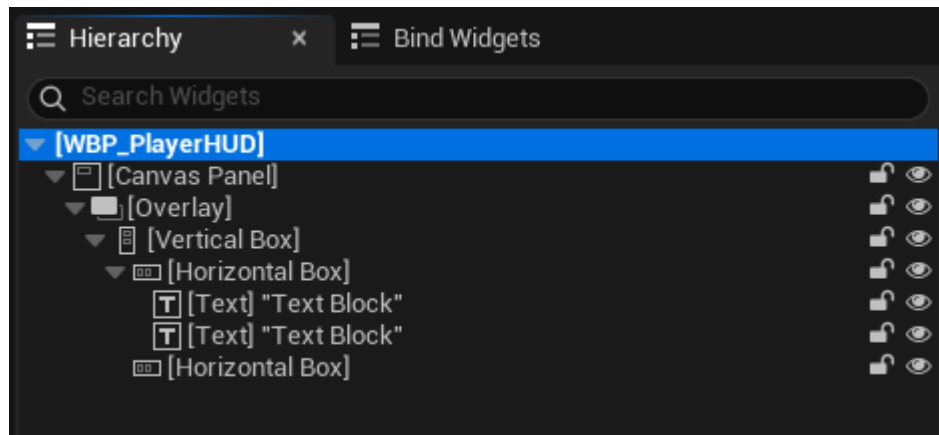
Desarrollo

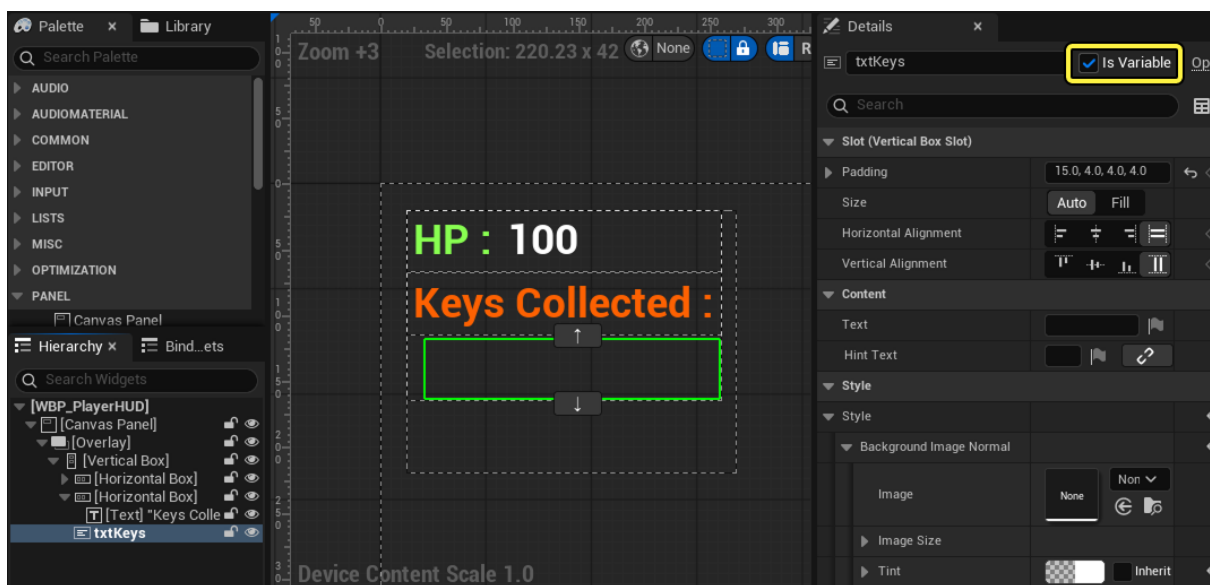
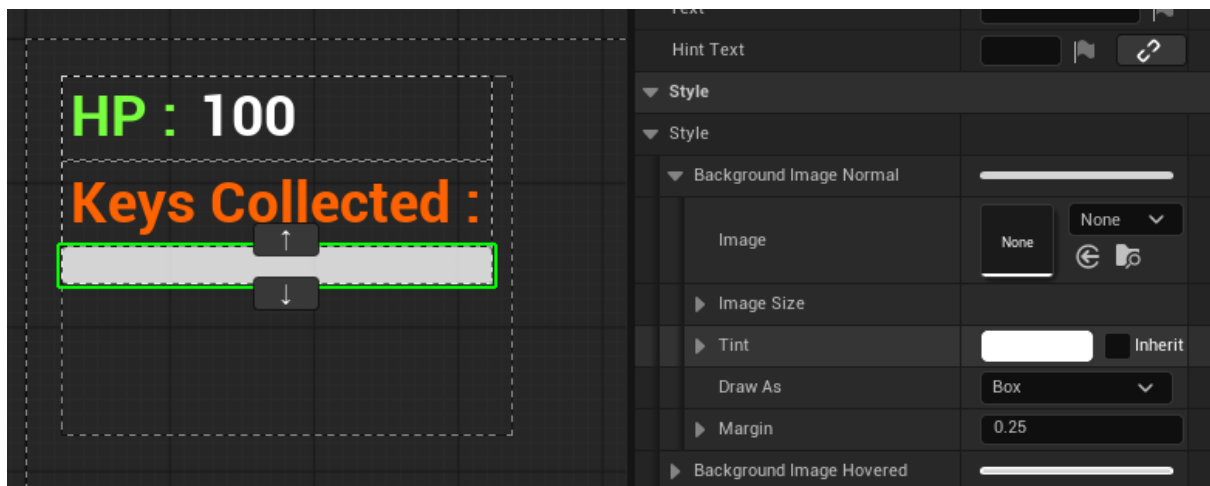
Se crea un widget que contiene elementos básicos como textos o imágenes. Se colocan dentro del canvas y se ajustan anclas y posiciones. Luego se crea un Blueprint de HUD que agrega el widget al viewport. Finalmente se prueba la interfaz en el juego.











Resultados esperados

Al completar la actividad, el estudiante será capaz de diseñar e implementar un widget funcional en Unreal Engine utilizando el sistema UMG. Se espera que comprenda cómo estructurar elementos visuales dentro del canvas, aplicar anclas correctamente y preparar componentes para diferentes resoluciones. Además, el estudiante deberá ser capaz de vincular el widget con el HUD del juego mediante Blueprints, logrando que la interfaz se muestre y actualice durante la ejecución. Esto permitirá sentar bases sólidas para crear menús, indicadores y sistemas de información interactivos.

3.2 Practica experimental 11

Cargo:	Docente Especialista		
Nombre:	Alexis Urgilés		
Asignatura:	Diseño de Videojuegos		
Carrera:	Alexis Urgilés	Nivel:	Tercer Nivel
Estudiante:			

ACTIVIDAD PRÁCTICO EXPERIMENTAL EN EL ENTORNO ACADÉMICO

Creación de un contador funcional mediante Blueprints y Widgets

Objetivos

Implementar un contador que aumente o disminuya durante el juego usando Blueprints.

Antecedentes/Escenario

Los contadores son fundamentales en la interfaz y la lógica de muchos juegos.

Recursos necesarios

UMG y Blueprints.

Planteamiento del problema

En la creación de un proyecto de videojuegos es imprescindible tener la capacidad de configurar un contador interactivo que se actualice en tiempo real.

Pasos por realizar

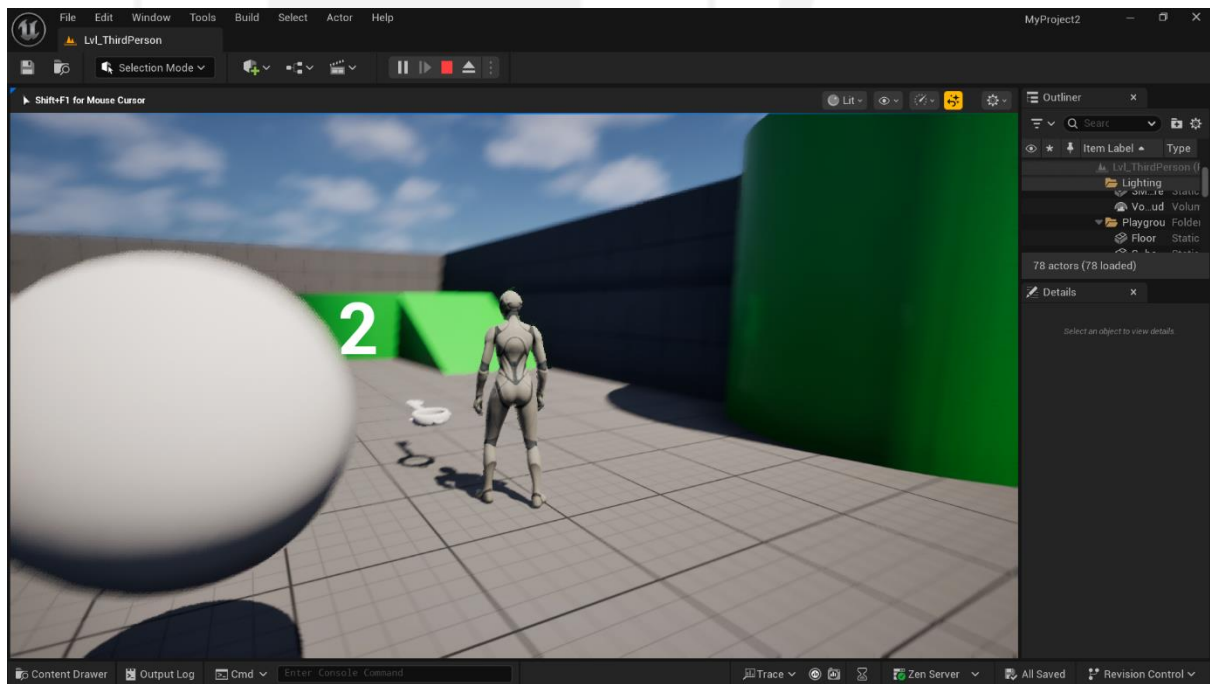
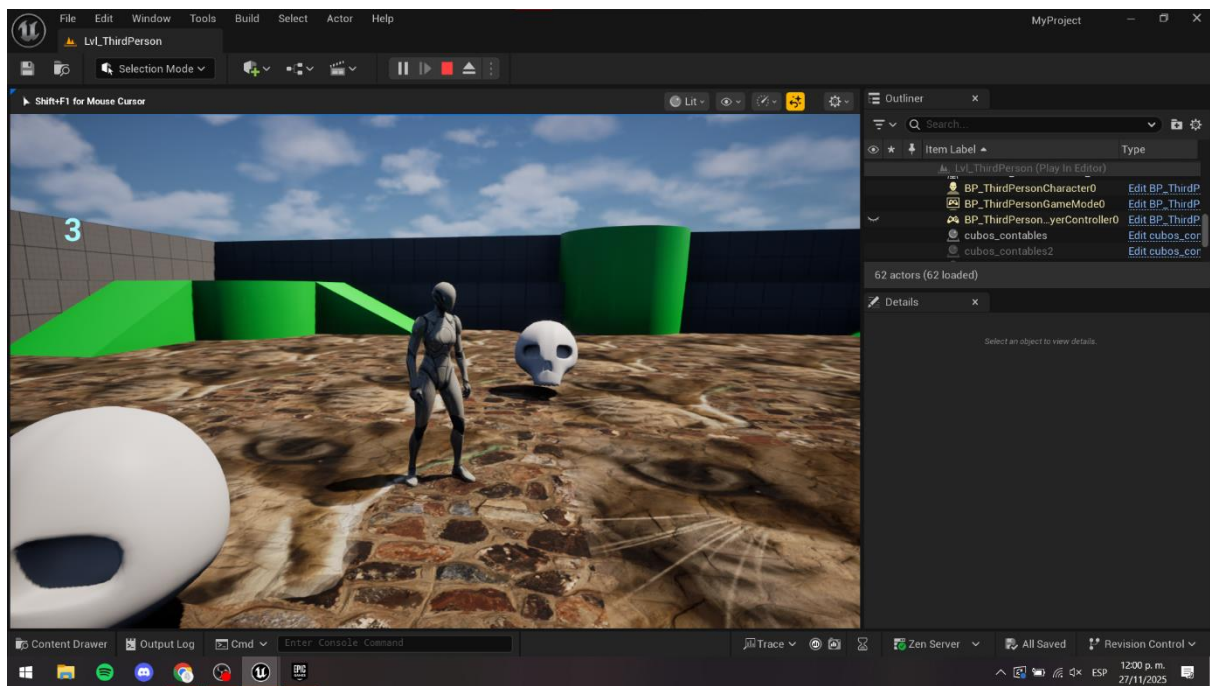
Paso 1: Crear variable en Blueprint.

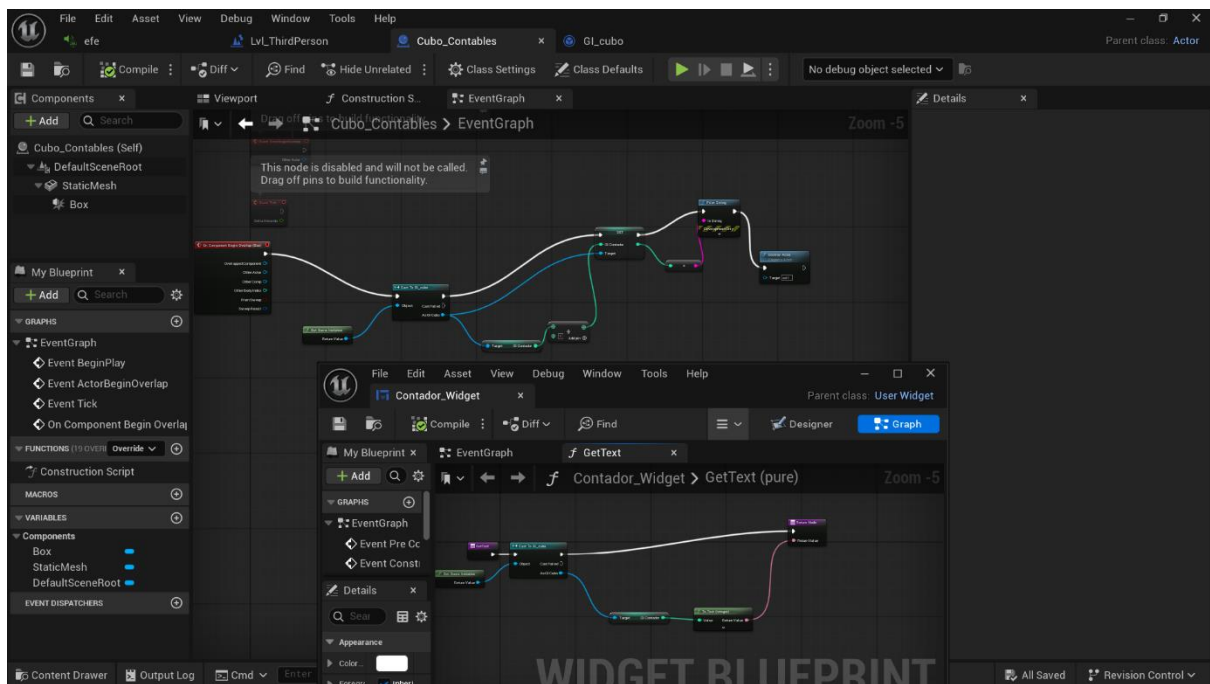
Paso 2: Conectarla al widget.

Paso 3: Actualizar valor mediante eventos.

Desarrollo

Se crea una variable numérica en el Blueprint principal del jugador. Se vincula al widget mediante binding o funciones. Durante la ejecución se incrementa o disminuye mediante eventos específicos como colisiones o temporizadores. El valor se ve reflejado en pantalla en tiempo real.





Resultados esperados

Al finalizar la actividad, el estudiante será capaz de implementar un contador totalmente funcional utilizando Blueprints y Widgets en Unreal Engine. Se espera que comprenda cómo declarar y gestionar variables, establecer una comunicación efectiva entre el Blueprint del jugador y la interfaz UMG, y actualizar valores en tiempo real mediante eventos del juego. Además, el estudiante demostrará dominio en la sincronización visual del contador, asegurando que los cambios producidos por interacciones, colisiones o temporizadores se reflejen con precisión en pantalla, fortaleciendo así la lógica base de sistemas interactivos.

4 UNIDAD 5: Mecánicas y Teaser

4.1 Practica experimental 12

Cargo:	Docente Especialista		
Nombre:	Alexis Urgilés		
Asignatura:	Diseño de Videojuegos		
Carrera:	Alexis Urgilés	Nivel:	Tercer Nivel
Estudiante:			

ACTIVIDAD PRÁCTICO EXPERIMENTAL EN EL ENTORNO ACADÉMICO

Creación de un video corto mostrando avances del proyecto

Objetivos

Generar un video breve que muestre las funcionalidades actuales del proyecto de videojuego.

Antecedentes/Escenario

La documentación audiovisual ayuda a presentar el estado del proyecto, comunicar avances y evidenciar el progreso técnico y visual.

Recursos necesarios

Grabador de pantalla.

Editor de video básico.

Planteamiento del problema

Registrar y presentar un video con claridad y coherencia.

Pasos por realizar

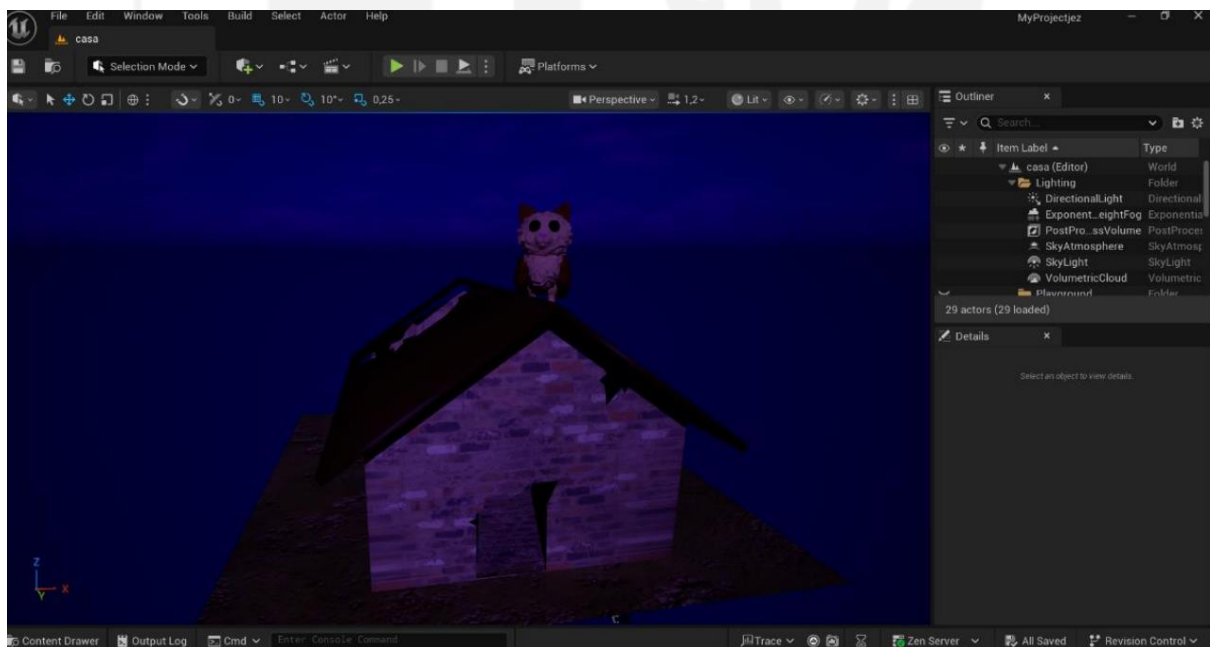
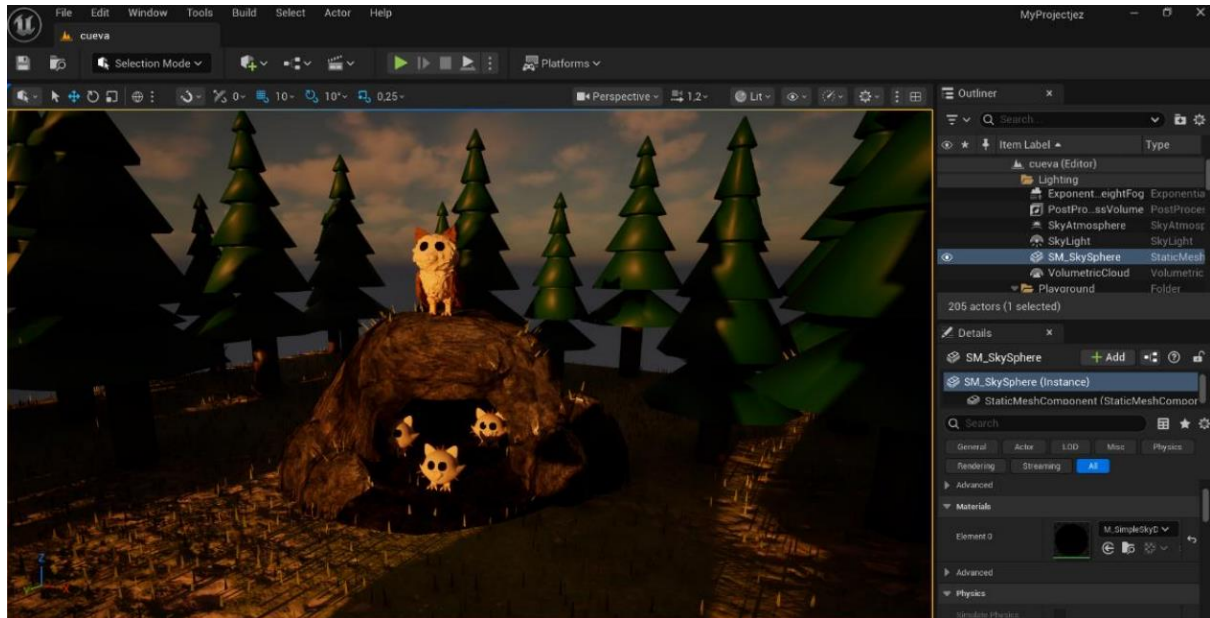
Paso 1: Grabar gameplay.

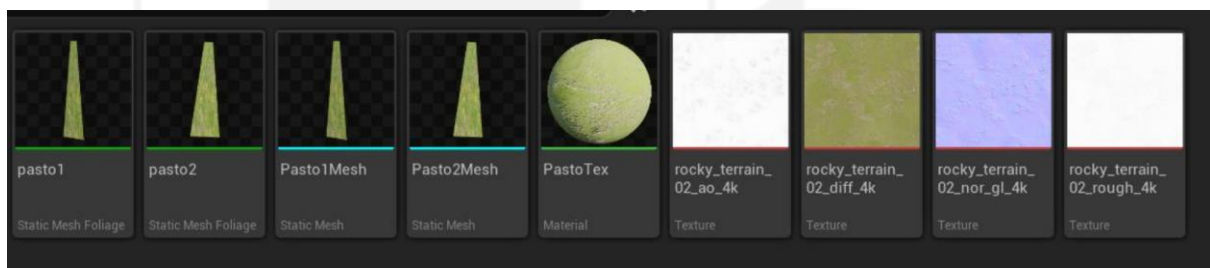
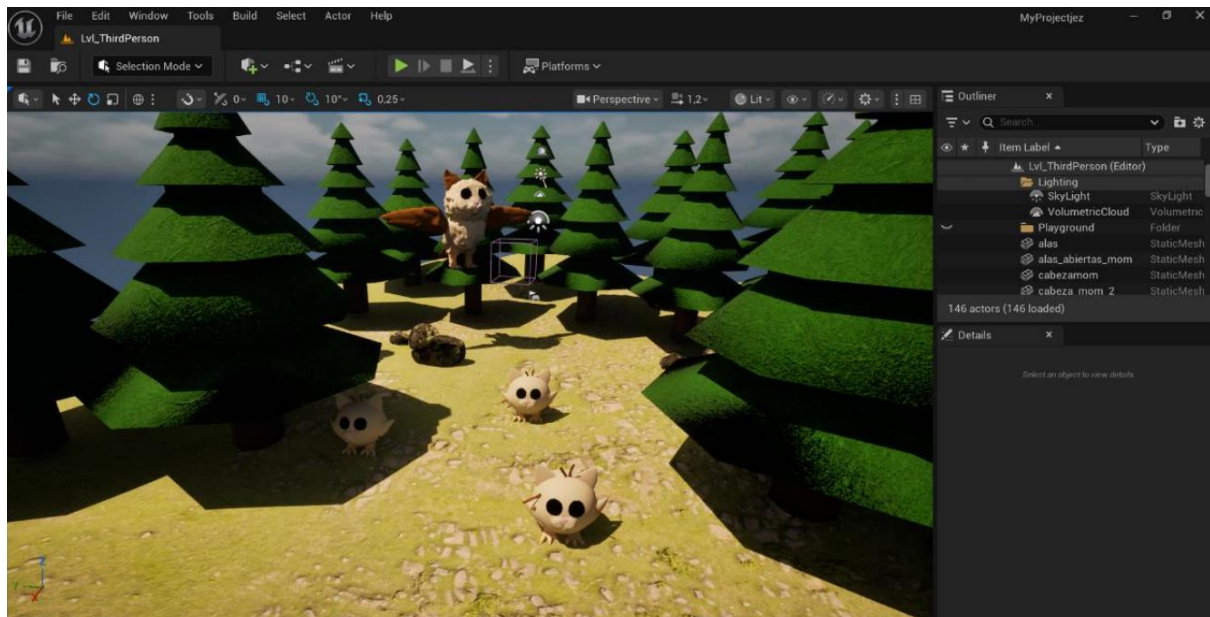
Paso 2: Editar clips.

Paso 3: Exportar video final.

Desarrollo

Se graban escenas que muestren interactividad, animaciones, UI y acciones principales. Se recortan clips para mantener fluidez y coherencia. Finalmente se exporta un video breve en formato mp4 que resume el progreso del proyecto.





Resultados esperados

Como resultado de esta actividad, el estudiante producirá un video corto y claro que muestre de manera efectiva el avance actual de su proyecto de videojuego. Se espera que el material audiovisual evidencie funcionalidades clave como interacción, animaciones, prototipos jugables, elementos de UI y cualquier mecánica implementada hasta el momento. El estudiante demostrará capacidad para capturar gameplay de forma estable, seleccionar los fragmentos más relevantes y editarlos para construir una presentación coherente y profesional. El video final permitirá evaluar visualmente el progreso técnico y creativo del proyecto.



4.2 Practica experimental 13

Cargo:	Docente Especialista		
Nombre:	Alexis Urgilés		
Asignatura:	Diseño de Videojuegos		
Carrera:	Alexis Urgilés	Nivel:	Tercer Nivel
Estudiante:			

ACTIVIDAD PRÁCTICO EXPERIMENTAL EN EL ENTORNO ACADÉMICO

Creación de un teaser inicial utilizando Sequencer

Objetivos

Crear un teaser breve con herramientas de cinematografía digital mediante Sequencer.

Antecedentes/Escenario

El sistema Sequencer permite crear cámaras, animaciones, transiciones y cortes como en una producción audiovisual.

Recursos necesarios

Sequencer en Unreal Engine.

Escena del proyecto.

Planteamiento del problema

Construir un teaser que presente el concepto del juego.

Pasos por realizar

Paso 1: Crear secuencia.

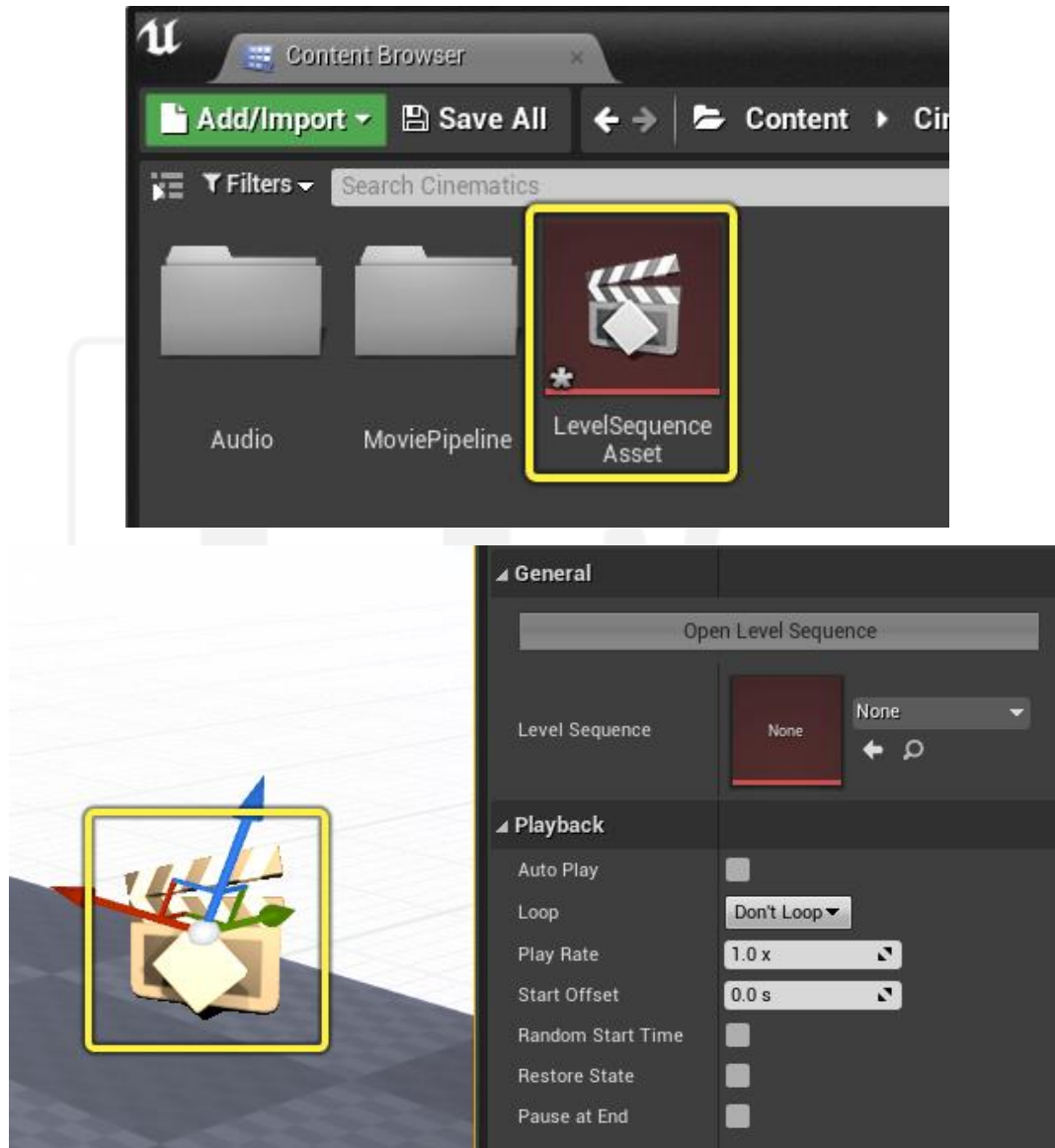
Paso 2: Añadir cámaras.

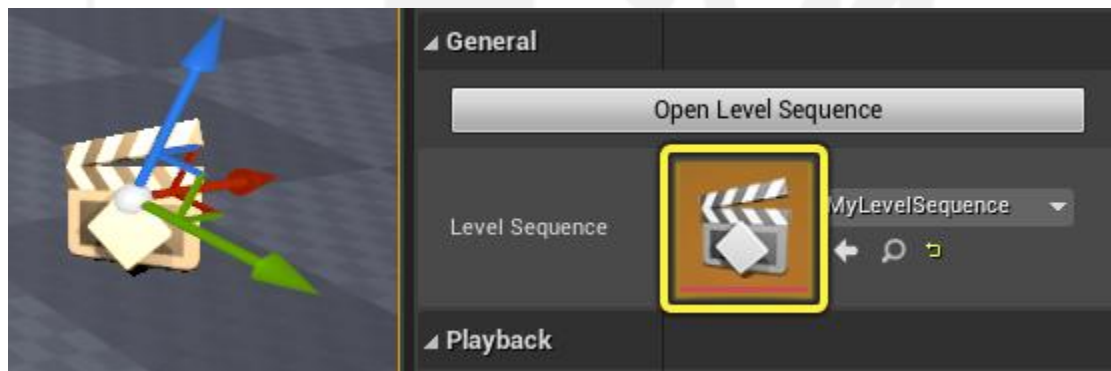
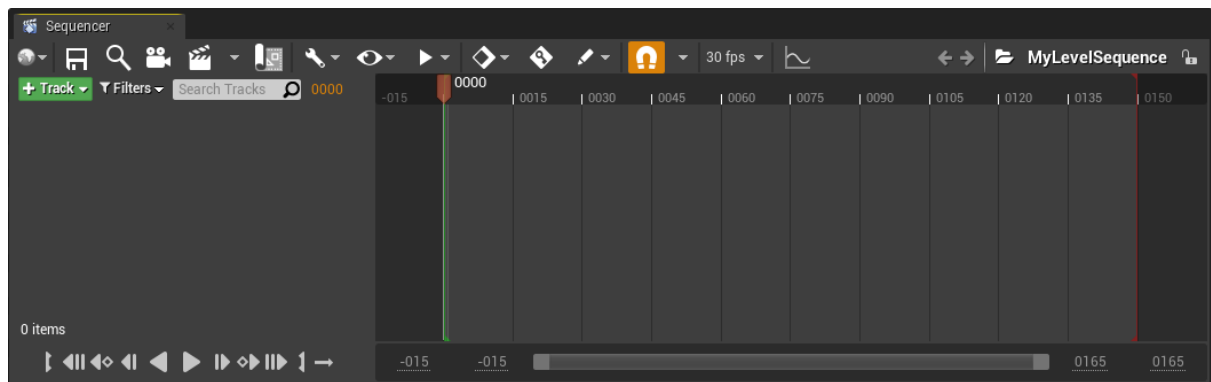
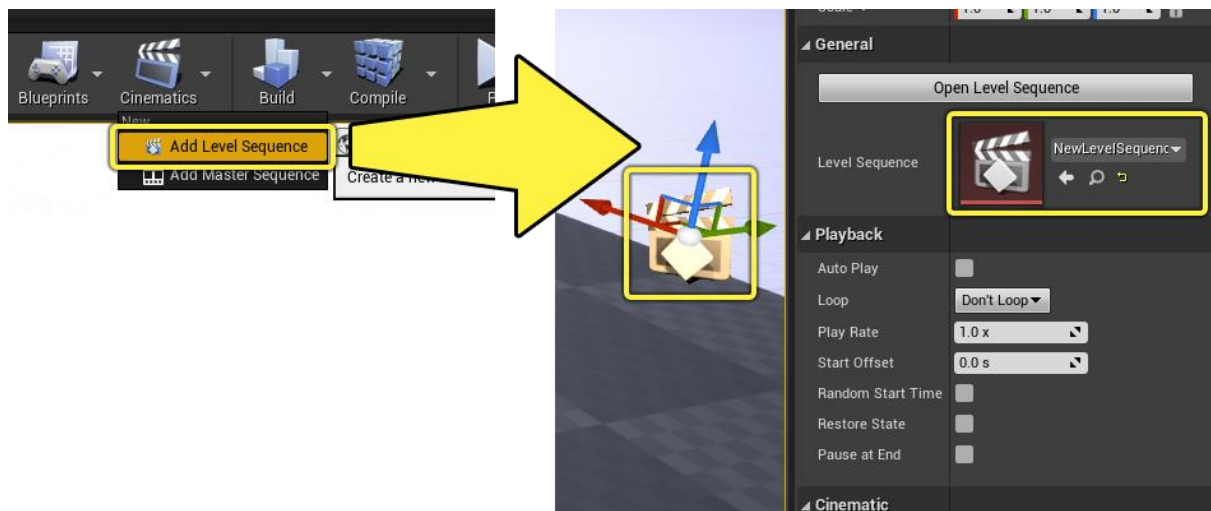
Paso 3: Animar elementos.

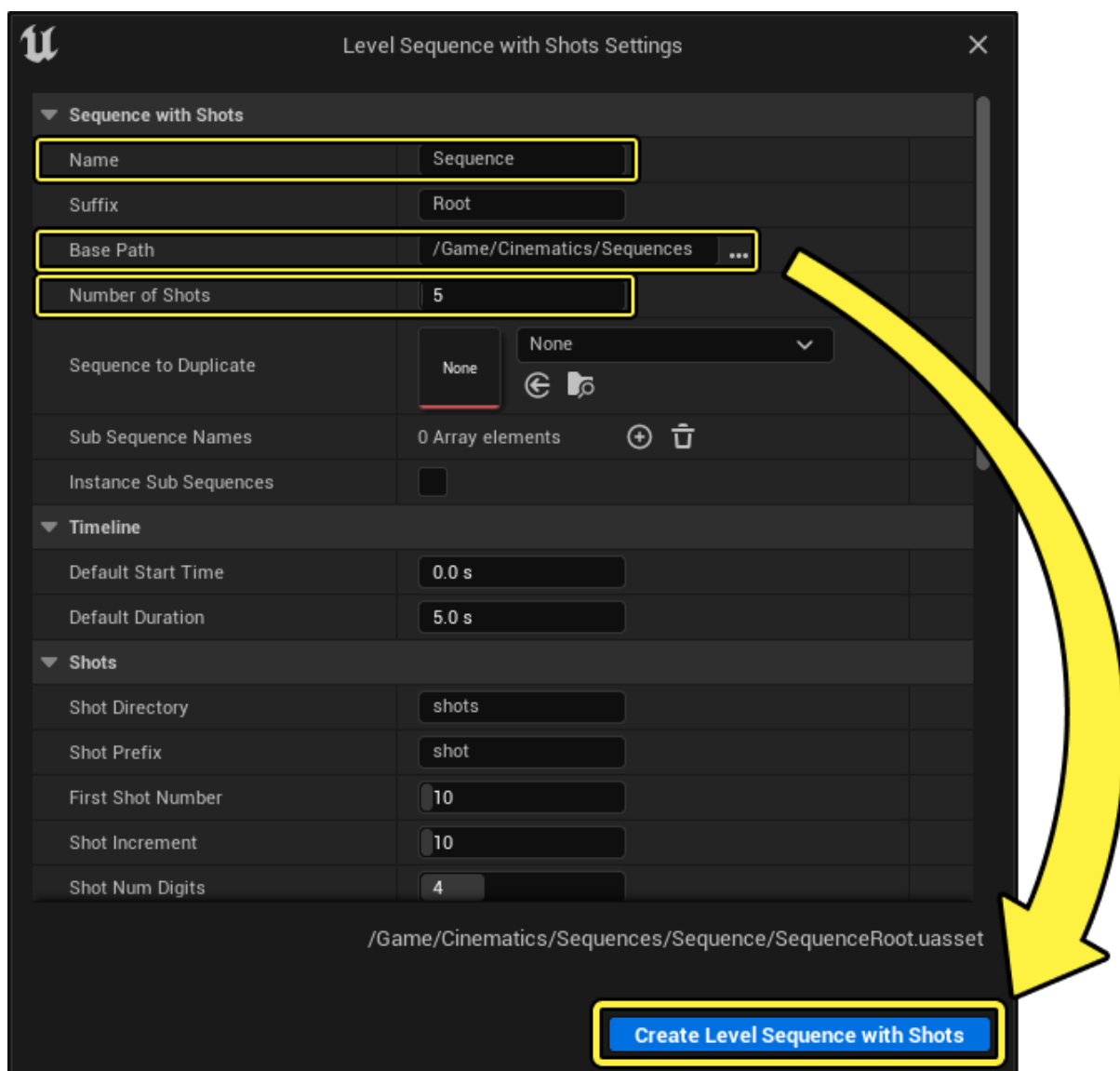
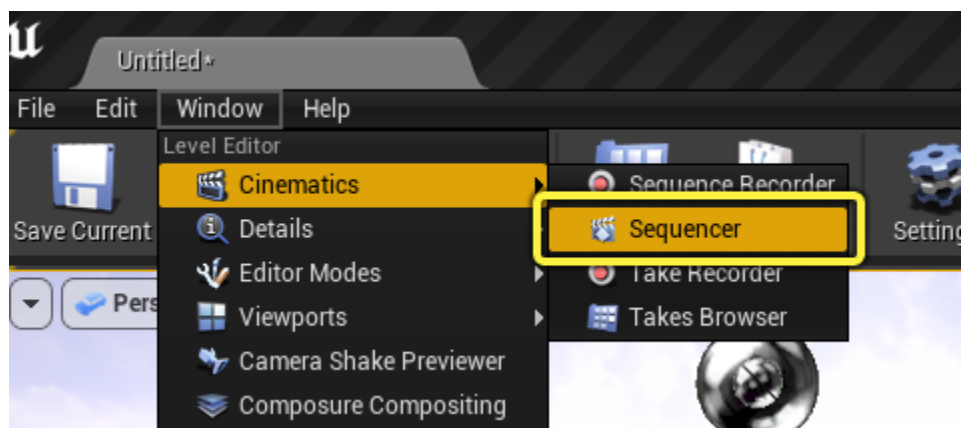
Paso 4: Exportar teaser.

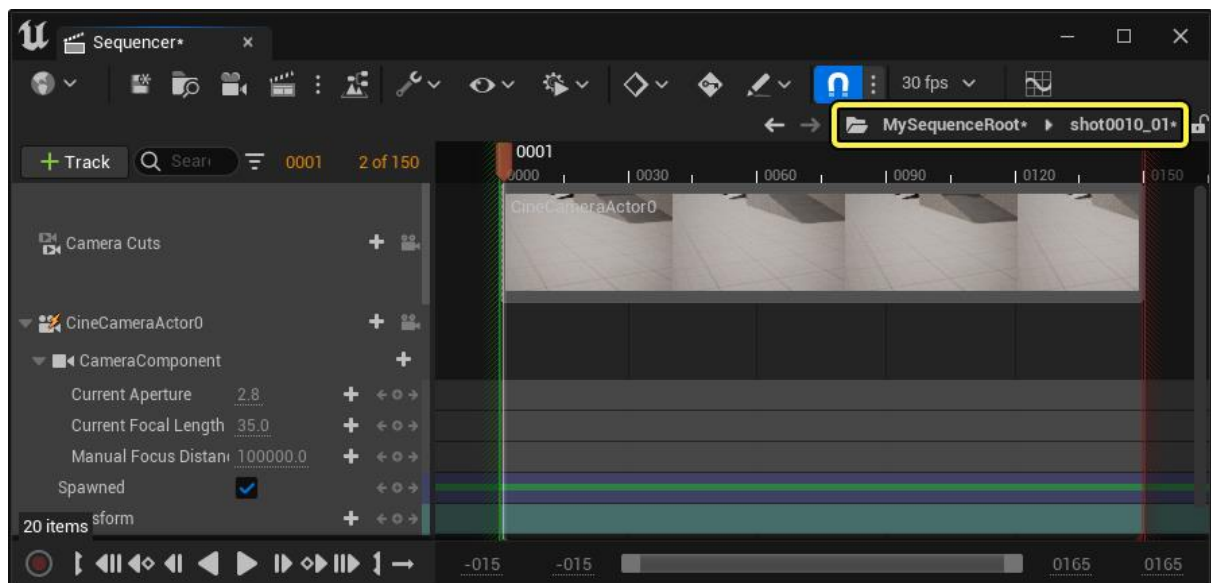
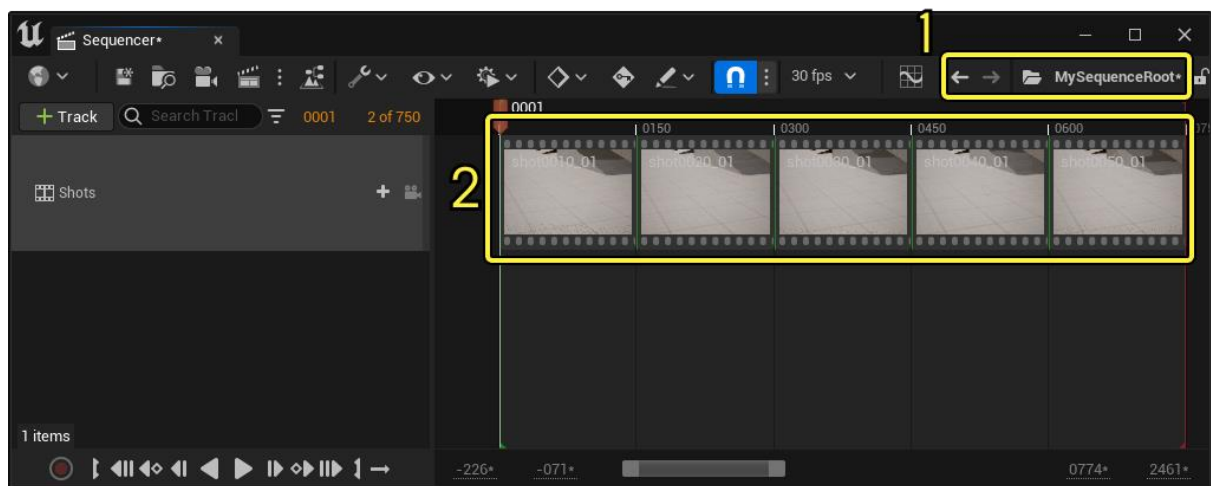
Desarrollo

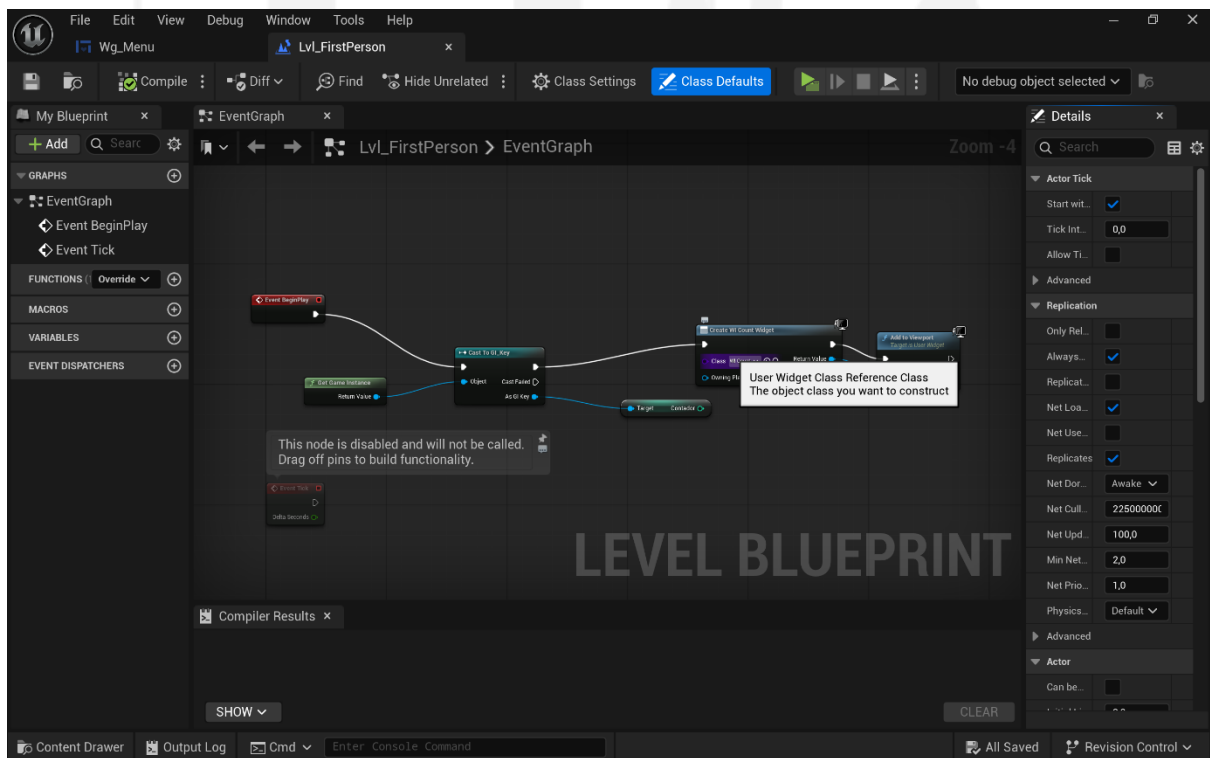
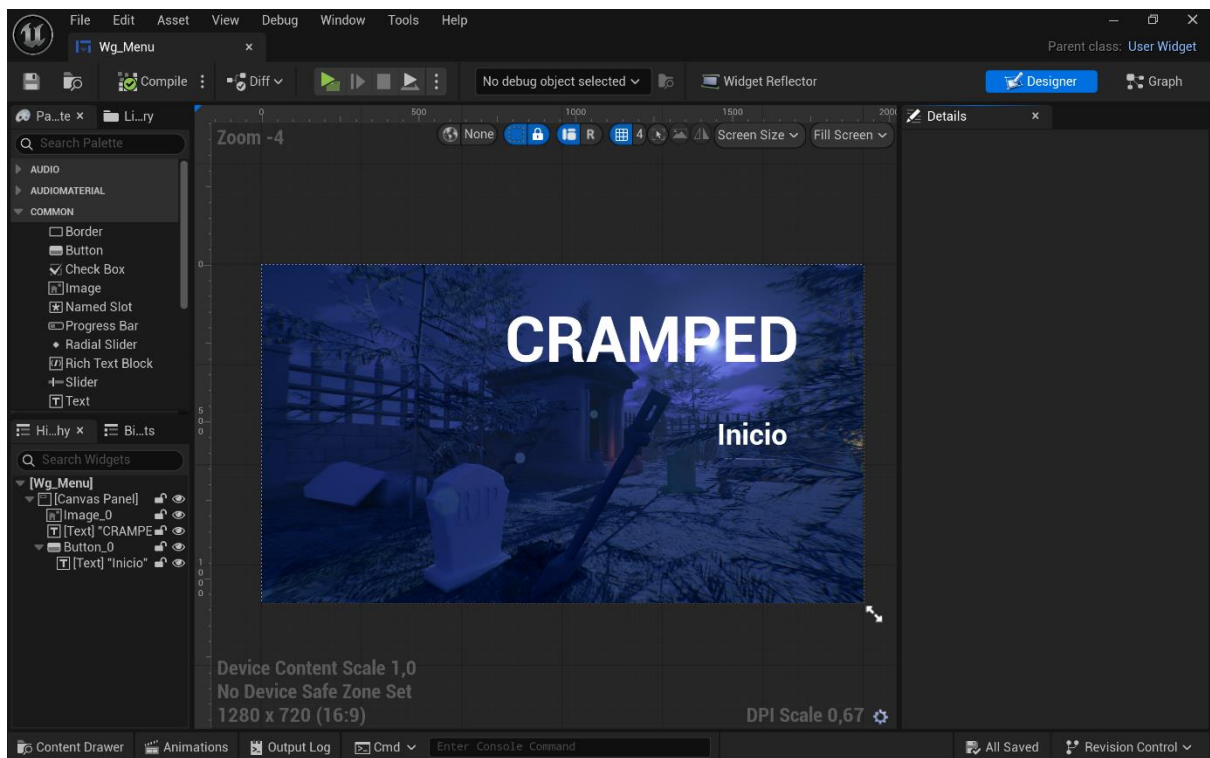
Se crea un nuevo nivel de secuencia. Se colocan cámaras con diferentes ángulos para capturar zonas destacadas del proyecto. Se aplican animaciones simples y transiciones. Finalmente se renderiza el video desde Rendermovie y se guarda como teaser inicial del juego.

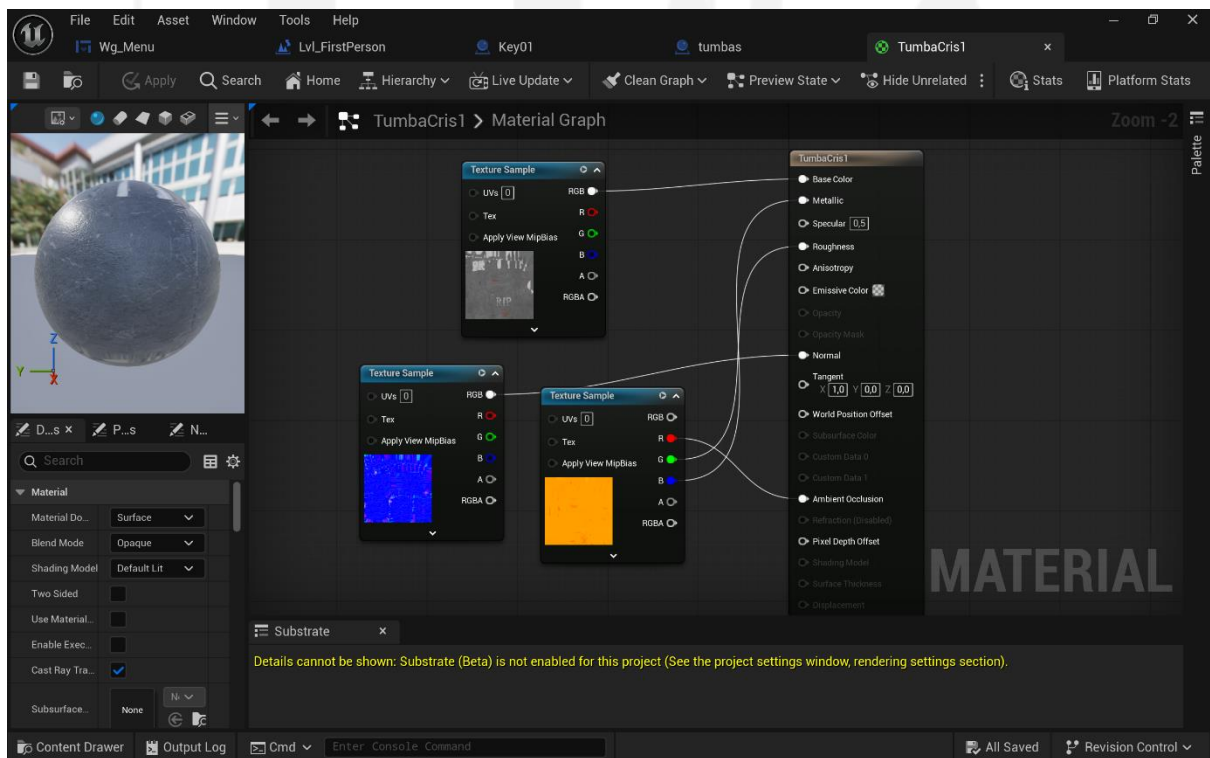
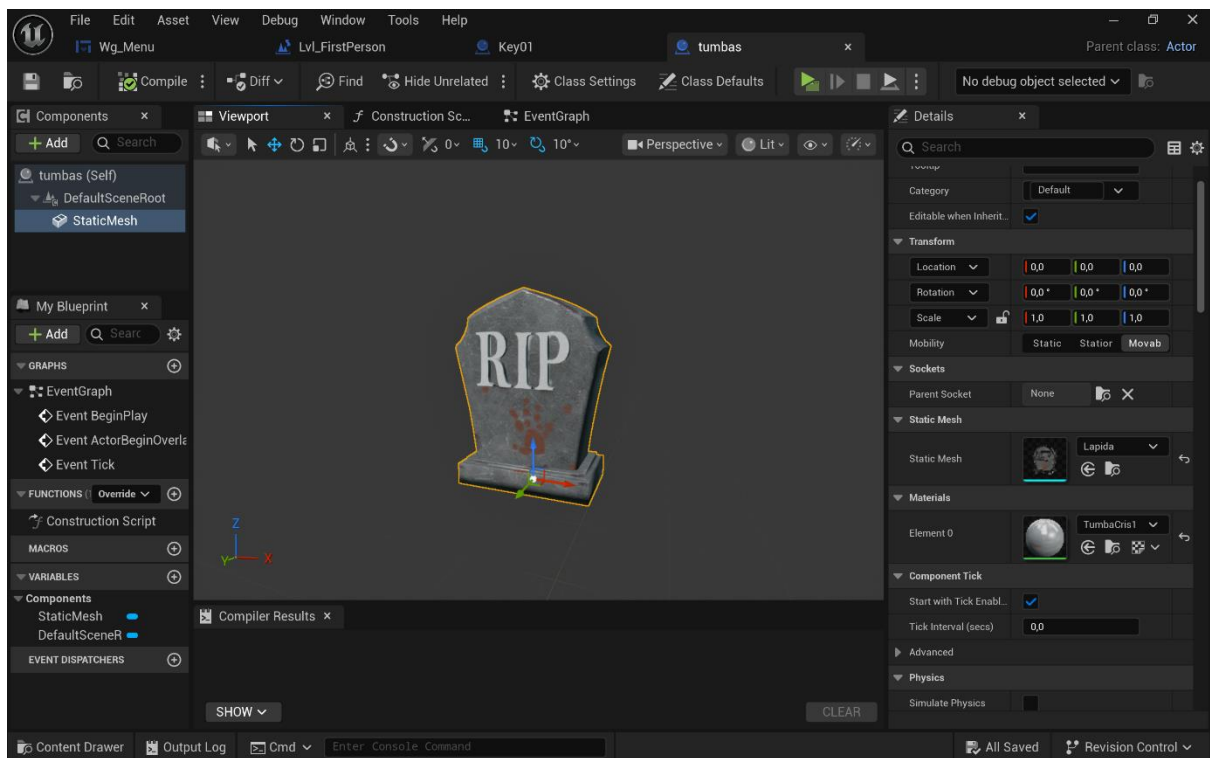


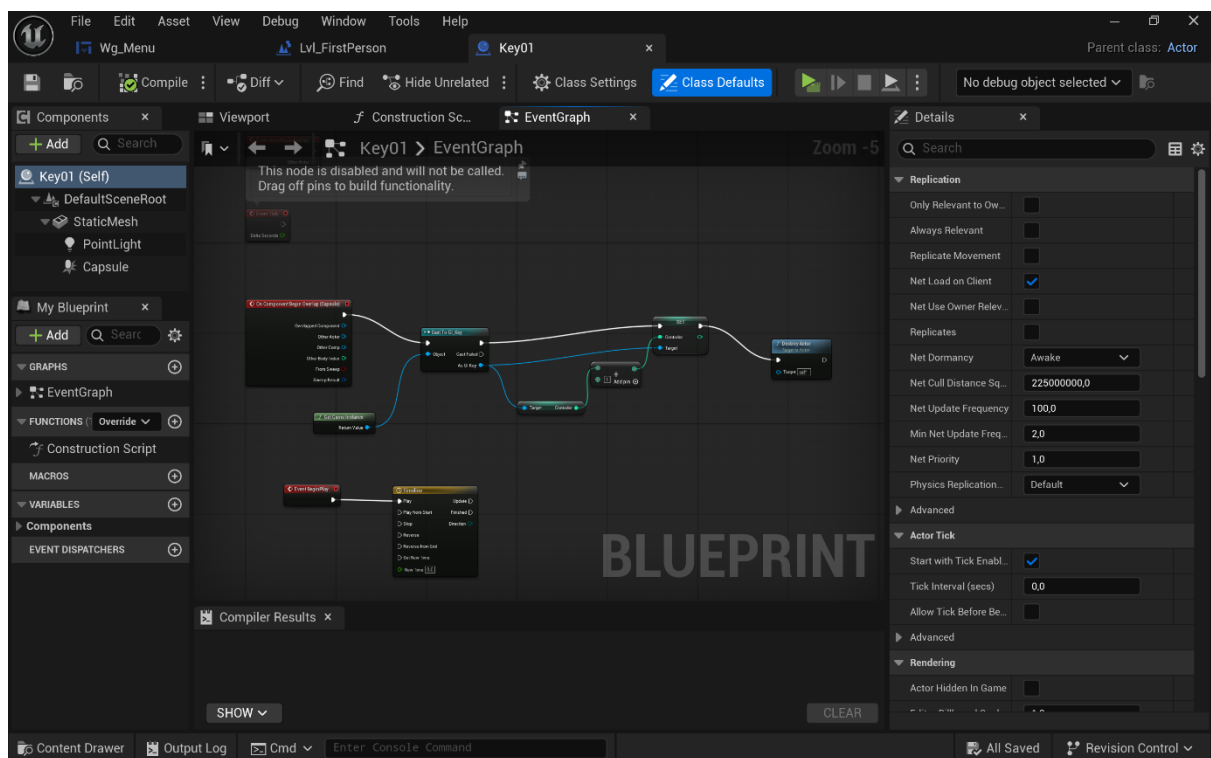


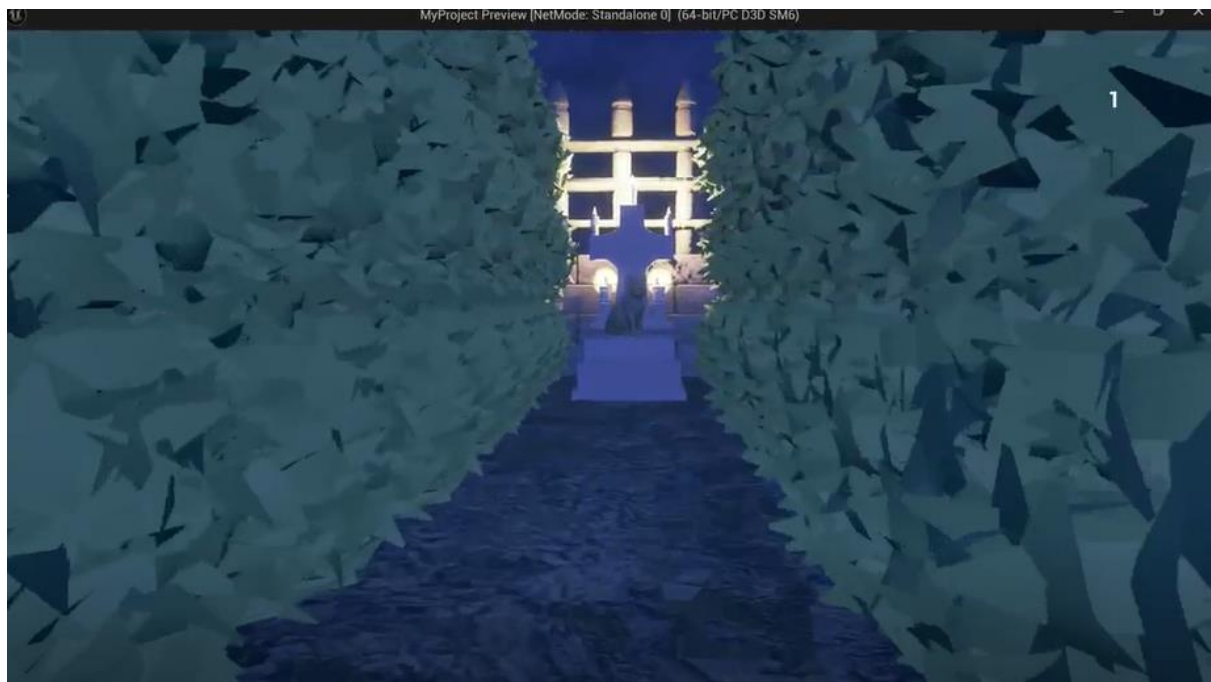














Resultados esperados

Al finalizar la actividad, el estudiante obtendrá un teaser inicial funcional que presentará de manera cinematográfica el concepto de su videojuego. Se espera que el teaser utilice adecuadamente cámaras, encuadres, movimientos y transiciones generados con Sequencer, demostrando comprensión de principios básicos de narración visual. El resultado debe mostrar una composición clara, con animaciones o elementos destacados del proyecto, permitiendo comunicar la atmósfera, estilo y premisa del juego. Asimismo, el estudiante evidenciará dominio inicial en la creación de secuencias y en la exportación del material final en formato de video.