



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

Guía general de  
**DISEÑO DE  
VIDEOJUEGOS**



ISBN: 978-9942-562-05-0



9 789942 562050



## GUÍA GENERAL DE DISEÑO DE VIDEOJUEGOS

**AUTOR: ALEXIS URGILÉS**

**EDICIÓN: 1**

**AÑO: 2024**

**TRABAJO EN EDICIÓN:**



**EQUIPO EDITORIAL:**

**ING KEYERMAN TOAPANTA**

**ING GABRIEL HOYOS**

Este material está protegido por derechos de autor. Queda estrictamente prohibida la reproducción total o parcial de esta obra en cualquier medio sin la autorización escrita de los autores y el equipo editorial. El incumplimiento de esta prohibición puede conllevar sanciones

establecidas en las leyes de Ecuador.

Todos los derechos están reservados.

**ISBN: 978-9942-562-05-0**





## **SOBRE EL AUTOR**



Alexis Xavier Urgilés Pillalaza. Adquirió un interés profundo en el campo audiovisual a una temprana edad, obteniendo apoyo de su familia incursionó en varias artes: pintura, música y danza andina. Consiguió un bachillerato de Físico Matemático y más tarde adquirió una licenciatura en Producción Audiovisual y Multimedia. Cuenta con más de 7 años creando contenido audiovisual. Posee experiencia en fotografía, ilustración digital, animación 3D y desarrollo de videojuegos.



## CONTENIDO

GUÍA GENERAL DE DISEÑO DE VIDEOJUEGOS.....	6
1. DESCRIPCIÓN DE LA ASIGNATURA .....	6
2. BIBLIOGRAFÍA .....	6
2.1. Básica .....	6
2.2. Complementaria.....	7
3. COMPETENCIAS GENÉRICAS Y ESPECÍFICAS .....	8
4. OBJETIVO GENERAL .....	9
5. FORMACIÓN CIUDADANA, VALORES Y HABILIDADES BLANDAS.....	10
6. NORMAS DE CLASE .....	10
7. SISTEMA DE EVALUACIÓN .....	11
8. UNIDADES .....	12
UNIDAD 1 BASES DEL DISEÑO DE VIDEOJUEGOS .....	12
Temas y Subtemas.....	12
TEMA 1: INTRODUCCIÓN AL DISEÑO DE VIDEOJUEGOS .....	12
<b>TEMA 2: NOCIONES DEL LENGUAJE DE PROGRAMACIÓN</b> .....	17
<b>TEMA 3: PRODUCTOS INTERACTIVOS</b> .....	18
<b>TEMA 4: LENGUAJE DE PROGRAMACIÓN C++</b> .....	19
Autoevaluación 1.....	21
Resumen de la Unidad 1.....	22
UNIDAD 2 PLANIFICACIÓN DE PROYECTOS .....	22
Temas y Subtemas.....	22
<b>TEMA 1: GAME DESIGNER</b> .....	22
<b>TEMA 2: NARRATIVA</b> .....	24
<b>TEMA 3: PROPUESTA ESTÉTICA</b> .....	25
<b>TEMA 4: DISEÑO DE PERSONAJES</b> .....	28
<b>TEMA 5: ASSETS</b> .....	35
Autoevaluación 2.....	38
Resumen de la Unidad 2.....	39
UNIDAD 3 RIGG Y SUS CONTROLADORES.....	40
Temas y Subtemas.....	40
TEMA 1: RIGGING .....	40
TEMA 2: CONTROLADORES .....	42
TEMA 3: PESOS .....	43
Autoevaluación 3.....	45

Resumen de la Unidad 3.....	46
UNIDAD 4 PROGRAMACIÓN EN UNREAL ENGINE.....	46
Temas y Subtemas.....	46
TEMA 1: INTRODUCCIÓN DE LA PLATAFORMA.....	46
TEMA 2: LEVEL BLUEPRINT Y BLUEPRINT CLASS.....	47
TEMA 3: PROGRAMACIÓN DEL PERSONAJE.....	49
TEMA 4: INTERFAZ DE USUARIO.....	51
Autoevaluación 4.....	53
Resumen de la Unidad 4.....	53
UNIDAD 5 TESTING-TEASER.....	54
Temas y Subtemas.....	54
TEMA 1: EMPAQUETADO.....	54
TEMA 2: TESTING.....	55
TEMA 3: TEASER.....	55
Autoevaluación 5.....	57
Resumen de la Unidad 5.....	58

## ÍNDICE DE FIGURAS

<b>Figura 1</b> Contenido UE.....	13
<b>Figura 2</b> ViewPort UE.....	14
<b>Figura 3</b> Plataforma Unity.....	15
<b>Figura 4</b> Splash Art.....	16
<b>Figura 5</b> Inicio de Mudbox.....	17
<b>Figura 6</b> Productos Interactivos.....	19
<b>Figura 7</b> C++.....	20
<b>Figura 8</b> Videojuegos.....	23
<b>Figura 9</b> Narrativa.....	24
<b>Figura 10</b> Propuesta Objetos.....	26
<b>Figura 11</b> Boceteado.....	28
<b>Figura 12</b> Modelado en Mudbox.....	29
<b>Figura 13</b> Propuesta Objetos.....	30
<b>Figura 14</b> Ventana de extracción de mapas.....	31
<b>Figura 15</b> Funcionamiento de mapas.....	32
<b>Figura 16</b> Retopología.....	33
<b>Figura 17</b> Herramientas de esculpido.....	34
<b>Figura 18</b> Mapa UV.....	35
<b>Figura 19</b> Herramientas Mudbox.....	36
<b>Figura 20</b> Herramientas textura.....	37



<b>Figura 21</b>	Rig .....	41
<b>Figura 22</b>	Herramientas de Rig .....	41
<b>Figura 23</b>	Herramientas .....	42
<b>Figura 24</b>	Controladores .....	43
<b>Figura 25</b>	WeightPainting .....	44
<b>Figura 26</b>	Pantalla de carga .....	47
<b>Figura 27</b>	Funcionamiento de mapas .....	49
<b>Figura 28</b>	Programación del ThirdpersonCharacter .....	50
<b>Figura 29</b>	Event BeginPlay .....	52
<b>Figura 30</b>	Trigger .....	52
<b>Figura 31</b>	Packaging .....	55
<b>Figura 32</b>	Sequencer .....	57





## GUÍA GENERAL DE DISEÑO DE VIDEOJUEGOS

### 1. DESCRIPCIÓN DE LA ASIGNATURA

Esta asignatura está centrada en la preproducción y producción de videojuegos encargada de establecer sistemáticamente la visión que tendrá un proyecto puesto que para su diseño es imprescindible que una idea sea funcional sin dejar de lado las necesidades del mercado. La materia aborda tres aspectos principales: la definición de estética, historia y jugabilidad de un proyecto. Por esta razón, es necesario implementar nociones en las áreas de programación, diseño de espacios, modelado orgánico e inorgánico, luz, textura, animaciones y físicas de elementos.

Los programas que serán revisados en esta asignatura son herramientas estándar en los procesos de trabajo de empresas que lideran el desarrollo de videojuegos a nivel mundial. Además, la formación del alumno está enfocada en el análisis crítico de las ideas que proyectan para que sus proyectos sean ejecutables y formen parte de su formación como profesionales.

### 2. BIBLIOGRAFÍA

#### 2.1. Básica

- Corbal, J. A. (2018). *Estética en videojuegos*: ( ed.). RA-MA Editorial.  
<https://elibro.net/es/lc/itq/titulos/106505>

“Estética en videojuegos” es una referencia bibliográfica que radica en su capacidad para proporcionar a los estudiantes una comprensión profunda y crítica sobre la estética. A través de este libro, los lectores pueden explorar cómo los elementos visuales, sonoros y narrativos se entrelazan para crear experiencias inmersivas y significativas en los videojuegos. Los estudiantes no solo pueden mejorar sus habilidades de diseño y desarrollo, sino también crear una apreciación más informada y reflexiva del medio.





- Corbal, J. A. (2017). *Curso de narrativa en videojuegos: ( ed.).* RA-MA Editorial. <https://elibro.net/es/lc/itq/titulos/106606>

“Curso de narrativa de videojuegos” es un texto que permite a los interesados en el diseño de videojuegos una comprensión profunda sobre la creación de historias dentro de este medio interactivo. A través de su análisis de la estructura narrativa, la integración de la jugabilidad y la creación de mundos inmersivos, el libro ofrece conocimientos indispensables para crear experiencias de juego envolventes y significativas. Además, al explorar cómo las decisiones del jugador influyen en la narrativa.

- García Arias, N. M. & Moreno Salamanca, Y. (2019). *Desarrollo de videojuegos: desde el diseño a la comercialización: ( ed.).* Ediciones de la U. <https://elibro.net/es/lc/itq/titulos/127124>

“Desarrollo de videojuegos: desde el diseño a la comercialización” Es un texto que maneja la conceptualización inicial hasta la estrategia de comercialización de un proyecto de videojuegos. Esto dentro de la materia ayuda a los estudiantes a comprender ciertos factores de importancia a tener en cuenta en la etapa de planificación de su objetivo como parte de un producto funcional.

- Cuadrado Alvarado, A. & Planells, A. J. (2020). *Ficción y videojuegos: teoría y práctica de la ludonarración: ( ed.).* Editorial UOC. <https://elibro.net/es/lc/itq/titulos/135287>

“Ficción y videojuegos: teoría y práctica de la ludo narración” Este texto es fundamental para los estudiantes de diseño de videojuegos que están interesados en la creación de un proyecto con una narrativa como parte de su mecánica dentro de su jugabilidad. Abordar la relación entre la narrativa y la jugabilidad en los videojuegos es necesario para establecer los eventos importantes que se deberá programar en las siguientes etapas. Además, los proyectos deberán cumplir con objetivos que estén ligados a un enfoque dentro de ciertas categorías para diferentes mercados y este texto ayudará a la clase en su expansión de criterio.

## 2.2. Complementaria.

- Contreras Espinosa, R. S. Eguía Gómez, J. L. & Lozano Muñoz, A. (2015). *Juegos multijugador: el poder de las redes en el entretenimiento: ( ed.).*







Editorial UOC. <https://elibro.net/es/lc/itq/titulos/57784>

El libro proporciona una perspectiva de otro tipo de proyecto diferente al que se está proyectando en clase. Los alumnos tendrán una herramienta más para poder aplicar su lógica de jugabilidad dentro de los videojuegos que se están diseñando.

- Parramón Paidotribo, E. (Il.). (2020). *Diseño y creación de personajes: desde el boceto hasta su realización en 3D*: (1 ed.). Parramón Paidotribo S.L. <https://elibro.net/en/lc/itq/titulos/248830>

Este texto brinda al estudiante conocimientos que complementan lo impartido en clase. La creación de un personaje desde su bocetaje a su etapa de modelado 3D es un tema muy extenso que siempre puede ser mejorado.

- Izquierdo, R. (2020). *Diseño emocional de personajes de videojuegos*: (1 ed.). RA-MA Editorial. <https://elibro.net/en/lc/itq/titulos/222653>

Este libro permite al estudiante obtener conocimientos más avanzados del diseño de caracterización de un personaje. Está ligado directamente a la narrativa y el guion del videojuego.

- Sánchez, T. E. & Mazuera, E. (2022). *Coloreando 3D en 2D + Nociones del color / Coloring 3D in 2D + Notions of Color*: (1 ed.). Bubok Publishing S.L. <https://elibro.net/es/lc/itq/titulos/260541>

Este texto da al estudiante un contexto más metódico de teorías de colores, formas y luces que son perfectamente aplicables en la creación de materiales y texturas en el proyecto trabajado.

### 3. COMPETENCIAS GENÉRICAS Y ESPECÍFICAS

El sector de la animación y de los videojuegos vive probablemente el mejor momento de su historia a nivel mundial, el uso de la animación como técnica para generar todo tipo de contenidos se ha extendido exponencialmente en los últimos años. la industria del entretenimiento (cine, televisión, música, plataformas digitales y videojuegos), está un paso





adelante en la concepción y realización de novedosos efectos visuales y animaciones digitales, que cada vez son más reales e impactantes.

El desarrollo de la tecnología ha mostrado que su influencia envuelve distintas áreas, esto se puede observar en la industria de videojuegos con mucha claridad. La creación de nuevos programas especializados y nuevas técnicas que mejoran tiempo y costos de producción. El mercado de la misma manera aumenta constantemente. En conclusión, la industria de los videojuegos no ha mostrado señales de declive y por ende nace la necesidad de la existencia de más profesionales en el área.

#### 4. OBJETIVO GENERAL

El objetivo de Diseño de Videojuegos es brindar a los estudiantes la capacidad de analizar y comprender los principios y fundamentos de programación de videojuegos, diseño de estética y jugabilidad. A través de la exploración crítica de morfología orgánica e inorgánica 3D, Rigging, DCC, programación en UE5, así como la aplicación de creación de guion, procesos de investigación y testing de videojuegos. Los estudiantes desarrollarán habilidades para analizar y resolver desafíos en plataformas de realidad virtual e inmersiva. Este conocimiento les permitirá aplicar programación por Blueprints de manera efectiva en la creación de proyectos que requieran procesos interactivos. Además, contribuir al desarrollo de un perfil profesional sólido y de esta manera solucionar problemas en la industria de plataformas digitales enfocadas en el diseño de videojuegos.



## 5. FORMACIÓN CIUDADANA, VALORES Y HABILIDADES BLANDAS

Educación ambiental: consumo y producción sostenibles

Valores & habilidades blandas: amor: comunicación asertiva y escucha activa

Valores & habilidades blandas: compromiso social: adaptabilidad

Valores & habilidades blandas: cultura: creatividad

Valores & habilidades blandas: gratitud: resiliencia

Valores & habilidades blandas: justicia: resolución de problemas

Valores & habilidades blandas: lealtad: liderazgo

Valores & habilidades blandas: optimismo: planificación y gestión del tiempo

Valores & habilidades blandas: orgullo nacional: pensamiento crítico

Valores & habilidades blandas: respeto: inteligencia emocional.

Valores & habilidades blandas: solidaridad: trabajo en equipo

Valores & habilidades blandas: tolerancia: flexibilidad

Valores & habilidades blandas: verdad y honradez: proactividad

## 6. NORMAS DE CLASE

En relación a las normas de clase, es importante destacar que la evaluación de los componentes de gestión académica se compone de tres notas sumativas, cada una con una puntuación máxima de 6.60/6.60, así como un proyecto práctico, como evaluación formativa que se valora con 3.40/3.40, lo que da un total de 10/10 para la calificación del módulo. Los parciales se califican en una escala de hasta 6.60 puntos, representando cada uno el 2.22 de la calificación total de 6.6 puntos. Para presentarse al proyecto final, el estudiante debe haber obtenido al menos 4.50 puntos sumando las tres primeras notas. En

caso de no alcanzar este mínimo en el proyecto, se otorga una oportunidad de recuperación dentro de las 48 horas laborables siguientes, según el calendario académico oficial. La nota mínima acumulada requerida para aprobar la asignatura es 7/10, y es esencial mantener al menos un 70% de asistencia a las clases. Los docentes deben informar a los estudiantes sobre sus notas individuales antes de registrarlas en el sistema, y se espera que los alumnos confirmen su aceptación y conformidad con estas calificaciones. Además, los docentes deben entregar un reporte de notas y asistencia a través del SGA y notificado a la coordinación de carrera y registrar las calificaciones en el sistema en un plazo máximo de 5 días posteriores a la recepción del proyecto final.

## 7. SISTEMA DE EVALUACIÓN

El sistema de evaluación de la asignatura se estructura para brindar una evaluación equilibrada y comprehensiva del desempeño de los estudiantes. La evaluación está compuesta por tres parciales, cada uno de los cuales contribuye con un 22% al resultado final, es decir 66% del total. Estos parciales evaluarán los conocimientos adquiridos durante distintas etapas del proceso de aprendizaje. El 34% restante proviene del Proyecto Final de la asignatura, el cual constituye una oportunidad para aplicar y demostrar habilidades y competencias en un contexto práctico. Este proyecto permite a los estudiantes explorar, analizar y abordar problemáticas relevantes en la asignatura, aplicando los conceptos y enfoques éticos adquiridos a lo largo del curso.





## 8. UNIDADES

### UNIDAD 1 BASES DEL DISEÑO DE VIDEOJUEGOS

#### Temas y Subtemas

**Tema 1: Introducción al Diseño de Videojuegos**

**Tema 2: Nociones del Lenguaje de Programación**

**Tema 3: Productos Interactivos**

**Tema 4: Lenguaje de Programación C++**

#### TEMA 1: INTRODUCCIÓN AL DISEÑO DE VIDEOJUEGOS

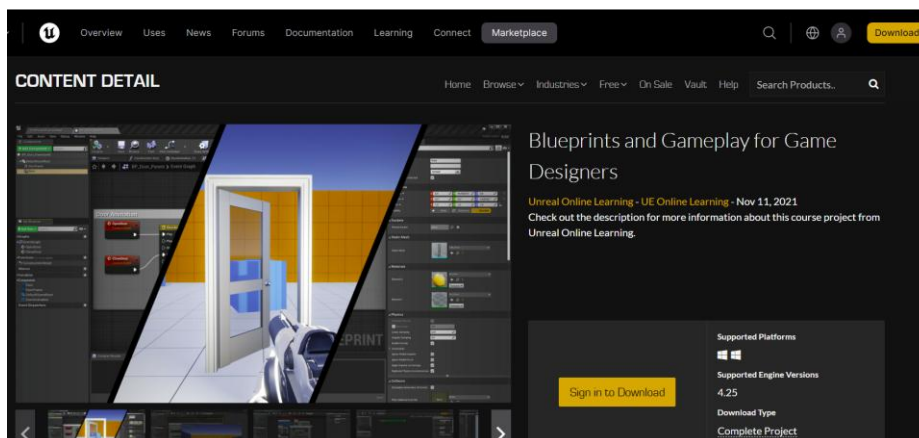
Esta unidad tiene como visión la presentación de conceptos básicos sobre la creación de videojuegos, se presentará lenguaje de programación C++ como introducción y un pequeño avance de su programación en Unreal Engine, pero se dará nociones del funcionamiento general de los lenguajes de programación. Se revisará el desarrollo que ha tenido UnrealEngine en la industria y cuáles son los programas que complementan el proceso de materialización de estas ideas. Además, se debe tener en consideración que el proceso de la creación de videojuegos está definido por una serie de pasos que conlleva la exploración de temas que forman parte de distintas especializaciones entre los cuales podemos mencionar: programador, diseñador



conceptual, guionista, ilustrador, escultor 3D, diseñador de interfaces, animador, colorista, texturizador, etc.

**Figura 11**

Contenido UE



Nota: La figura 1 muestra contenido que se proporciona en UE para el aprendizaje de diseño y desarrollo de videojuegos. Fuente: <https://www.unrealengine.com/marketplace/en-US/product/blueprints-and-gameplay-in-development>

El diseño de videojuegos constituye en gran parte al proceso previo a la materialización de ideas, por esta razón es imprescindible que exista una fase de lluvia de ideas en la cual se estable el objetivo que el proyecto va a cumplir. En esta unidad se expondrá la importancia de la planificación y la organización de los procesos que se realizarán.

La industria ha ido avanzando con pasos agigantados las últimas décadas y con esto el desarrollo de nuevos programas que realizan funciones especializadas ha ido aumentando. Es necesario adentrarse en el flujo del mercado que lidera los productos más aceptados en la actualidad. Los motores de desarrollo de videojuegos que lideran la industria son Unity y Unreal Engine cada uno con destrezas que los hacen únicos.

## ROLL DEL DISEÑADOR

La persona encargada de diseñar un proyecto de videojuego debe tener conocimientos de todas las fases por las cuales el proyecto atravesará y debe mantener un criterio bien formado para poder tomar decisiones que estén a la par de los objetivos establecidos para poder responder a las necesidades del mercado. En síntesis, el diseñador marca la pauta que el videojuego tendrá y debe mantener cierto nivel de conocimiento de varias áreas, pero no necesariamente debe especializarse en todas.

La planificación y el proceso de posproducción está liderada por el diseñador y está marcada por el cronograma subjetivo de cada líder de proyecto, pero es claro que los primeros pasos que se tomarán son la definición del objetivo del proyecto y el tipo de target al cual será dirigido. Este proceso da lugar a las primeras nociones de la propuesta estética y también la planificación de las mecánicas que del videojuego.

Dentro de esta unidad se explorará el trabajo que se puede adquirir mediante diferentes motores y para ello se revisará cuáles son los motores y herramientas que el mercado tiene como referente.

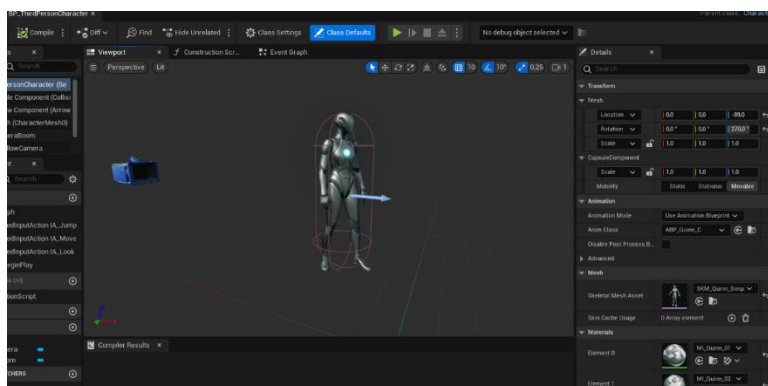
## DESARROLLO DE CONCEPTOS

### UNREAL ENGINE

Es un motor de videojuegos creado por la compañía Epic Games lanzado oficialmente en 1988 y su última versión estable fue lanzada el 6 de septiembre de 2023. Fue pensada como desarrollador de shooter en primera persona, pero al mantener un potente sistema de simulación en tiempo real ha permitido abarcar la industria del cine y televisión siendo uno de los precursores de XR (realidad extendida). Su alto grado de portabilidad está dado por su estructura de programación, con su código escrito en C++ ha desarrollado un sistema de programación más amigable para los desarrolladores. Los blueprints son uno de los factores que ha permitido que su motor sea más aceptado dentro de la comunidad de desarrolladores de videojuegos indies

**Figura 22**

*ViewPort UE*



Nota: En la figura 2 se observa como UE5 trabaja con elementos para la creación de videojuegos. Elaboración: Autor.

### Unity



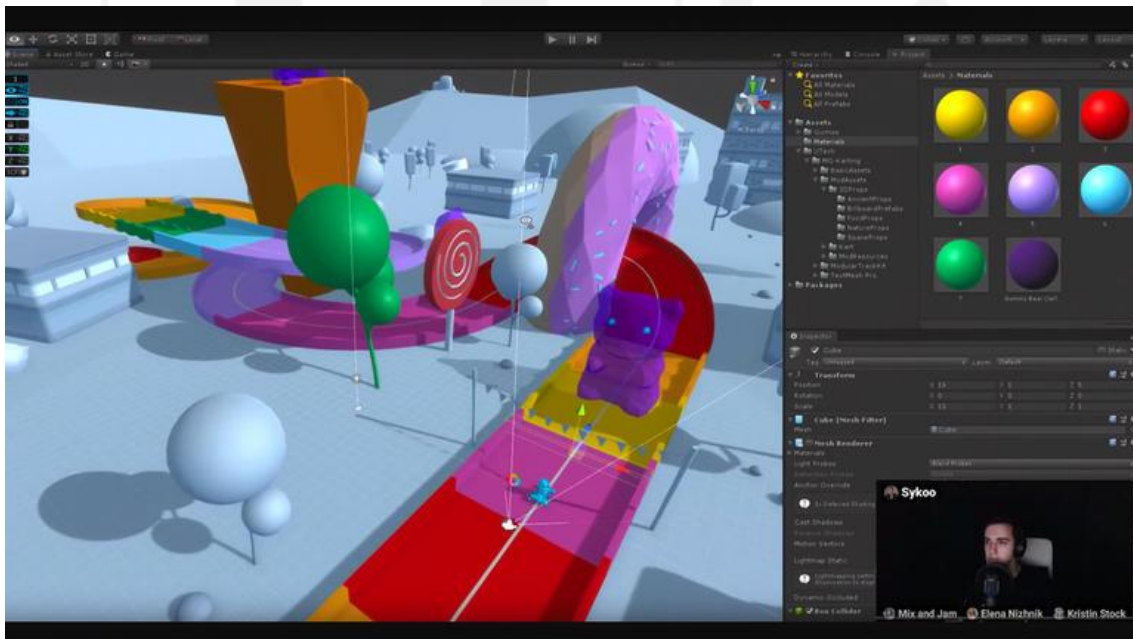


Es un motor de videojuegos creado por la compañía Unity Technologies lanzado oficialmente en 2005 como motor para juegos Mac OS X y su última versión estable fue lanzada el 15 de enero de 2023. Esta plataforma está centralizada en el funcionamiento de videojuegos 2D y 3D gracias a la implementación de Mecanim que apoya a la animación de movimientos fluidos de personajes con una interfaz eficiente. Y aunque es un motor que ha triunfado en el mercado sus polémicas por comisiones han hecho que haya un rechazo por la comunidad de desarrolladores en todo el mundo.

Para el diseño de videojuegos la plataforma, el motor y el código que usaremos para programar nuestras ideas no forma parte de todo el trabajo a realizar, los DCC (Digital Content Creation) o los assets que serán implementados en los proyectos forman parte de una línea de trabajo diferente. Para la creación de assets que serán implementados dentro del proyecto se debe incursionar en el estudio de diferentes programas de creación y edición de contenido digital.

**Figura 33**

*Plataforma Unity*



Nota: La figura 3 muestra la plataforma que Unity usa para trabajar un proyecto de videojuegos. Fuente: [https://unity.com/sites/default/files/styles/16\\_9\\_s\\_scale\\_width/public/2019-11/first-game-jam.jpg?itok=RTPzbvBS](https://unity.com/sites/default/files/styles/16_9_s_scale_width/public/2019-11/first-game-jam.jpg?itok=RTPzbvBS)

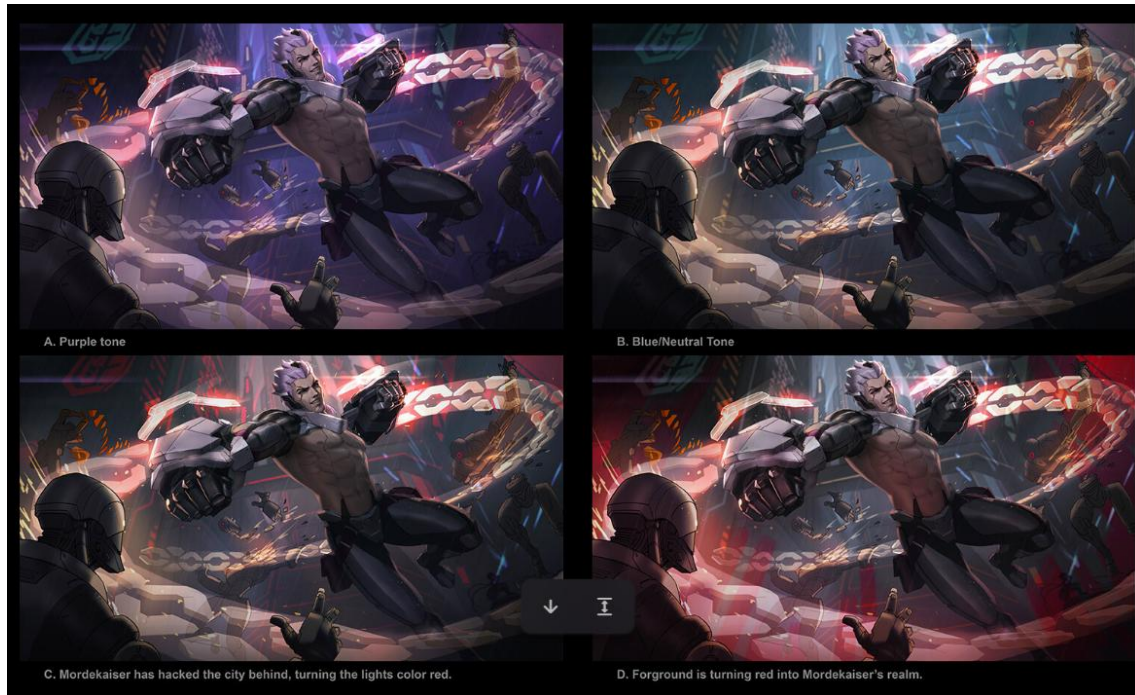
## Photoshop

Es un programa destinado principalmente para la edición de imágenes, pero su potencial ha permitido que sus funcionalidades influyan a la industria de la animación y los videojuegos, con

sus herramientas de edición y dibujo, este software permite realizar la propuesta de diseño de los personajes, objetos, locaciones e inclusive permite la creación de la propuesta de estética del proyecto. En etapas de modelado y texturizado esta herramienta permite la edición de UVs que son esenciales para el correcto despliegue gráfico.

#### Figura 44

#### Splash Art



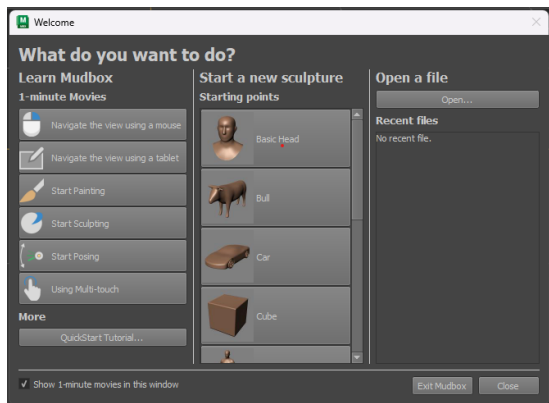
Nota: La figura 4 muestra proceso del splash art que está destinado para un videojuego. Fuente: <https://www.artstation.com/artwork/mDeAJE>

#### MUDBOX

Los modelados 3D que requieran ser esculpidos serán creados dentro de esta plataforma para poder tener un workflow adecuando con Maya al momento de creación de UVs o retopología de elementos. También dará pauta para la introducción y creación de materiales únicos.

**Figura 55**

*Inicio de Mudbox*



Nota: En la figura 5 se puede observar los diferentes tipos de proyectos que se pueden realizar en esta plataforma.  
Elaboración: Autor.

Otros programas que son parte del proceso de creación de videojuegos serán revisados, pero no profundizados tales como: Adobe Substance 3D, 3DS MAX, Illustrator, Premiere Pro, Houdini y Zbrush.

## **TEMA 2: NOCIONES DEL LENGUAJE DE PROGRAMACIÓN**

El diseñador de videojuegos debe mantener cierto nivel de entendimiento de cómo funciona el desarrollo de un videojuego para poder comprender factores esenciales de creación y ejecución de un proyecto. La introducción a las nociones esenciales de la programación es dada al inicio de la materia para que el proceso de entendimiento sea extendido, y a largo plazo su comprensión esté ligada al desarrollo del proyecto final.

Para esta unidad se enfocará en los conceptos básicos del lenguaje de programación utilizados en programación y se familiarizará con la sintaxis y la escritura del lenguaje C++. De esta manera se espera adquirir habilidades para escribir códigos de manera eficiente.

### **Lenguaje de Programación**

Un lenguaje de programación es un conjunto de reglas y símbolos utilizados para escribir programas informáticos. Estos programas son conjuntos de instrucciones que le dicen a una computadora qué hacer, cómo hacerlo y en qué orden hacerlo. Los lenguajes de programación actúan como intermediarios entre el ser humano y la máquina, permitiendo que las personas comuniquen sus ideas y soluciones a través de un código comprensible por la computadora.

Conceptos que se mencionarán:

- Sintaxis y Semántica
- Abstracción
- Portabilidad
- Paradigmas

### TEMA 3: PRODUCTOS INTERACTIVOS

Los productos interactivos mantienen una estrecha relación con los lenguajes de programación puesto que es fundamental en el proceso de creación y desarrollo de experiencias digitales. Su relación la podemos discernir de la siguiente manera

- **Implementación de la interactividad:** Los productos interactivos, como aplicaciones móviles, videojuegos, sitios web interactivos y aplicaciones de realidad aumentada/virtual, requieren interactividad para involucrar a los usuarios. Los lenguajes de programación permiten a los desarrolladores implementar esta interactividad al escribir código que responda a las acciones y entradas del usuario.
- **Desarrollo de funciones y características:** Los lenguajes de programación permiten a los desarrolladores crear funciones y características específicas que hacen que un producto interactivo sea único y funcional. Por ejemplo, en un videojuego, los lenguajes de programación se utilizan para crear mecánicas de juego, inteligencia artificial para personajes no jugadores (NPC), sistemas de física, efectos visuales y mucho más.
- **Personalización y adaptabilidad:** Los lenguajes de programación permiten la personalización y adaptabilidad de los productos interactivos para satisfacer las necesidades y preferencias de los usuarios. Mediante la escritura de código, los desarrolladores pueden implementar opciones de personalización, ajustes de configuración y características específicas solicitadas por los usuarios.
- **Optimización de rendimiento:** Los lenguajes de programación también juegan un papel importante en la optimización del rendimiento de los productos interactivos. Los desarrolladores pueden utilizar técnicas de programación eficientes y optimización de



código para garantizar que el producto funcione de manera fluida y responda rápidamente a las acciones del usuario, incluso en dispositivos con recursos limitados.

Los productos interactivos y los videojuegos están estrechamente relacionados debido a su naturaleza centrada en la experiencia del usuario y la participación activa del jugador. Los videojuegos son una forma específica de producto interactivo que ofrece una experiencia inmersiva y dinámica, donde los jugadores interactúan con el contenido del juego y toman decisiones que afectan la progresión y el resultado del mismo. Ambos tipos de productos se centran en la creación de experiencias atractivas y envolventes mediante el uso de elementos como la interactividad, la narrativa, los gráficos, el sonido y la jugabilidad. Además, comparten técnicas y tecnologías comunes en su desarrollo, como el diseño de interfaz de usuario, la programación de comportamientos interactivos, la optimización del rendimiento y la integración de contenido multimedia.

**Figura 66**

*Productos Interactivos*



Nota: La figura 6 muestra un usuario interactuando con un producto interactivo. Fuente: [https://www.elsotanoperdido.com/images/EntradillasExtra/Videojuegos\\_ENtra7.jpg](https://www.elsotanoperdido.com/images/EntradillasExtra/Videojuegos_ENtra7.jpg)

#### **TEMA 4: LENGUAJE DE PROGRAMACIÓN C++**

C++ es un lenguaje de programación de alto rendimiento, desarrollado como una extensión de C a principios de los años 80. Destaca por su equilibrio entre programación de bajo y alto nivel,



ofreciendo características como la orientación a objetos, la programación genérica y un manejo eficiente de la memoria. Utilizado ampliamente en la creación de sistemas operativos, aplicaciones de escritorio, videojuegos y software de bajo nivel, su sintaxis está influenciada por C, pero agrega funcionalidades adicionales como clases, herencia, plantillas y manejo de excepciones, lo que lo convierte en una herramienta poderosa pero también puede ser desafiante de dominar debido a su complejidad.

### C++ en los videojuegos

El lenguaje de programación C++ y los videojuegos están fuertemente relacionados debido a la capacidad de adaptarse al alto y bajo nivel. C++ se ha convertido en uno de los lenguajes más populares en la industria del desarrollo de videojuegos debido a su eficiencia en el uso de recursos, su capacidad para optimizar el rendimiento y su flexibilidad para implementar sistemas complejos. Los programadores de videojuegos utilizan C++ para crear los motores de juego, que son el núcleo de cualquier videojuego, así como para desarrollar sistemas de física, gráficos, inteligencia artificial, redes y otros aspectos fundamentales de los juegos. La combinación de la potencia de C++ con las bibliotecas y herramientas específicas para el desarrollo de videojuegos permite a los desarrolladores crear experiencias interactivas de alto nivel.

**Figura 77**

C++

```
#pragma once


#include "GameFramework/Actor.h"
#include "MyActor.generated.h"

UCLASS()
class AMyActor : public AActor
{
    GENERATED_BODY()

public:
    // Sets default values for this actor's properties
    AMyActor();

    // Called when the game starts or when spawned
    virtual void BeginPlay() override;

    // Called every frame
    virtual void Tick( float DeltaSeconds ) override;
};
```



**UNREAL  
ENGINE**

Nota: La figura 7 muestra la estructura en la cual trabaja UE5. Fuente: [https://docs.unrealengine.com/4.27/Images/ProgrammingAndScripting/ProgrammingWithCPP/IntroductionToCPP/image\\_0.webp](https://docs.unrealengine.com/4.27/Images/ProgrammingAndScripting/ProgrammingWithCPP/IntroductionToCPP/image_0.webp)

En esta materia se estudiará UnrealEngine para la creación de los proyectos, y este motor utiliza C++ como lenguaje de programación para desarrollar y extender el motor de juego, así como para crear la lógica del juego y manipular el contenido interactivo. Los desarrolladores de juegos pueden escribir código en C++ para crear y personalizar comportamientos de personajes, sistemas de juego, efectos visuales, sistemas de física, inteligencia artificial y más. Además, Unreal Engine proporciona una API robusta y una arquitectura modular que permite a los desarrolladores utilizar conceptos avanzados de programación, como herencia, polimorfismo, plantillas y diseño orientado a objetos, para crear juegos complejos y de alto rendimiento. Un ejemplo de cómo UE utiliza conceptos avanzados de programación en C++ es su sistema de componentes y sistemas, que permite la modularidad y reutilización del código al separar las funcionalidades del juego en componentes individuales. Los desarrolladores pueden crear y conectar componentes en C++ para definir el comportamiento de los objetos del juego de manera flexible y eficiente.

### Autoevaluación 1

1. ¿Cuáles son los conceptos fundamentales que se deben comprender al aprender un nuevo lenguaje de programación?
2. ¿Cómo se define una variable y cuál es su función en la programación?
3. ¿Qué es un producto interactivo?
4. ¿Qué lenguaje de programación usa UE5?
5. ¿Cómo se puede ajustar el brillo y el contraste de una imagen en Photoshop?
5. ¿Cuál es la diferencia entre guardar y exportar una imagen en Photoshop?
7. ¿Cuál es la diferencia entre la pintura 2D y la pintura 3D en Mudbox?
8. ¿Qué son las capas de pintura en Mudbox y cómo se utilizan para crear texturas complejas?
9. ¿Cuál es la función del panel de escena en Mudbox y cómo se utiliza para organizar elementos en un proyecto?
10. ¿Qué es un GameObject en Unity y qué tipos de componentes puede contener?





## Resumen de la Unidad 1

La unidad presenta conceptos básicos que permiten al estudiante crear procesos mentales lógicos que más tarde lo aplicarán en su programación con blueprints. Se establece el rol del diseñador dentro de un proyecto de video juegos y la importancia de definir correctamente un objetivo. Por último, se da nociones del lenguaje de programación C++.

## UNIDAD 2 PLANIFICACIÓN DE PROYECTOS

### Temas y Subtemas

**Tema 1: Game Designer**

**Tema 2: Narrativa**

**Tema 3: Propuesta Estética**

**Tema 4: Diseño de Personajes**

**Tema 5: Assets**

### TEMA 1: GAME DESIGNER

Es imprescindible entender el rol de Diseñador de videojuegos, sus funciones y operaciones a cargo. La unidad tiene como prioridad explicar la importancia del diseño de videojuegos antes de su etapa de desarrollo, para esto se exponen casos reales de la industria en los cuales los proyectos no han podido salir a flote por su falta de profesionalismo o la ejecución de ideas obsoletas en sus proyectos.



**Figura 88**

*Videojuegos*

**El MMO de 'League of Legends' llevaba tres años en desarrollo. Riot Games ha confirmado que lo han reiniciado**

- Riot Games busca hacer un MMO diferente a las propuestas disponibles actualmente en el mercado
- Tras este anuncio, la compañía nos prepara para una larga temporada de silencio

1 comentario    



Nota: La figura 8 muestra el desempeño inadecuado de la dirección de un proyecto a cargo de una empresa líder.

Fuente: <https://www.xataka.com/videojuegos/mmo-league-of-legends-llevaba-tres-anos-desarrollo-riot-games-ha-confirmado-que-han-reiniciado>

Las funciones que mantiene un Diseñador de videojuegos están ligadas a las etapas de moldeamiento y ejecución por esta razón están conectadas directamente a la posproducción y producción de un proyecto, la etapa de preproducción no mantiene un interés muy focal pero aún mantiene cierta influencia.

Dentro de la posproducción se establece en primer plano el concepto, la narrativa y la idea principal, esto también puede nacer a partir de una mecánica o jugabilidad que este pensada como punto característico y de allí partir a las siguientes etapas.

La concepción de la idea principal debe estar guiada por el objetivo y su afán por responder las necesidades del mercado. En esta etapa se manifiestan las lluvias de ideas y se concreta el core target. También el diseñador se encarga de manejar y laborar la propuesta estética y por último dentro de la posproducción se manejan los bocetos de los personajes y escenarios que serán digitalizados.

En la producción entran procesos de desarrollo y acabado del proyecto, se digitaliza todas las ideas previamente establecidas. Este proceso está monitoreado por el diseñador por esta razón los desarrolladores, ilustradores, animadores y creadores de contenido multimedia responden directamente a él.

## TEMA 2: NARRATIVA

El diseñador está encargado de entender como la historia guiará las bases de un proyecto teniendo en consideración una narrativa que sea fiel a las necesidades del mercado actual. El guion narrativo dentro de los videojuegos procura contar una historia mediante elementos acciones y diseños.

### Diseño Narrativo

Forma parte de cada elemento perceptible e imperceptible que afecta en algún grado al storytelling del proyecto. La narrativa en esta industria usa recursos nacidos en esta área, pero también aplica recursos de otros medios como cine, teatro, televisión y literatura. La jugabilidad dentro del proyecto también cuenta como parte de la narrativa. Las características que definen las mecánicas de un juego son una forma más de contar una historia o transmitir un mensaje. También tenemos a los videojuegos que están conectados a una narrativa previa de otro medio de comunicación y sirve como complemento o base para entender el contexto de una historia.

### Figura 99

*Narrativa*



Nota: La figura 9 muestra un videojuego que ha mantenido la narrativa y la mecánica de la mano(Darksouls). Fuente: [https://img.redbull.com/images/c\\_crop,x\\_0,y\\_0,h\\_1055,w\\_2344/c\\_fill,w\\_1700,h\\_756/q\\_auto,f\\_auto/redbullcom/2018/01/17/9e235bdc-0636-4131-a8f3-91a7446899fb/dark\\_souls\\_nintendo-switch](https://img.redbull.com/images/c_crop,x_0,y_0,h_1055,w_2344/c_fill,w_1700,h_756/q_auto,f_auto/redbullcom/2018/01/17/9e235bdc-0636-4131-a8f3-91a7446899fb/dark_souls_nintendo-switch)

En la serie de animación creada por Riot “ARCANE” el guion está basado en las historias creadas para el videojuego League of Legends. Un ejemplo contrario está en la serie de videojuegos nacidos a partir de la popularidad de las películas de Harry Potter las cuales fueron basadas en

obras literarias. De esta manera, se puede entender que una buena base de diseño narrativo no limita a que un guion permanezca en un solo medio. También existe la narrativa en los juegos como parte de la jugabilidad del proyecto como lo aplica el director Hidetaka Miyazaki o Hideo Kojima. Las historias como recompensa dentro de las mecánicas o los fragmentos activan el interés en el guion y esto se puede estimular con los diálogos de los NPC o historias que suceden en segundo plano.

La escritura del guion difiere de las demás industrias puesto que el papel de la escritura y su habilidad para definir una historia no es imprescindible, a no ser que el punto importante del juego sea su narrativa. El encargado del guion no debe establecerse necesariamente por una historia sólida, pero sí por elementos y momentos que definan la historias. Estos momentos serán clave para las acciones que realizará el usuario y de esta forma tener un avance en la narrativa.

En proyectos grandes diseñadores y directores están a cargo de presentar la noción principal que espera que el proyecto tenga y con esto los guionistas pueden desarrollar los primeros borradores. Los métodos de trabajo de la industria difieren, pero podemos organizar las acciones principales que deben tomarse a consideración.

- Definición de la historia Principal
- Definición del mundo ordinario
- Definición de Personajes y motivos
- Listar posible desarrollo de diálogos.
- Revisión y Ajustes

El proceso de creación de un guion para videojuegos es un ejercicio multidisciplinario que requiere la integración de elementos narrativos, de diseño y técnicos para dar vida a una experiencia interactiva y envolvente. Desde la conceptualización de la historia y los personajes hasta el diseño de la jugabilidad y los escenarios, cada paso del proceso implica una cuidadosa planificación y colaboración entre escritores, diseñadores, artistas y programadores. El resultado es un guion que no solo guía al jugador a través de una aventura, sino que también establece la base para la creación de un mundo virtual rico y lleno de posibilidades.

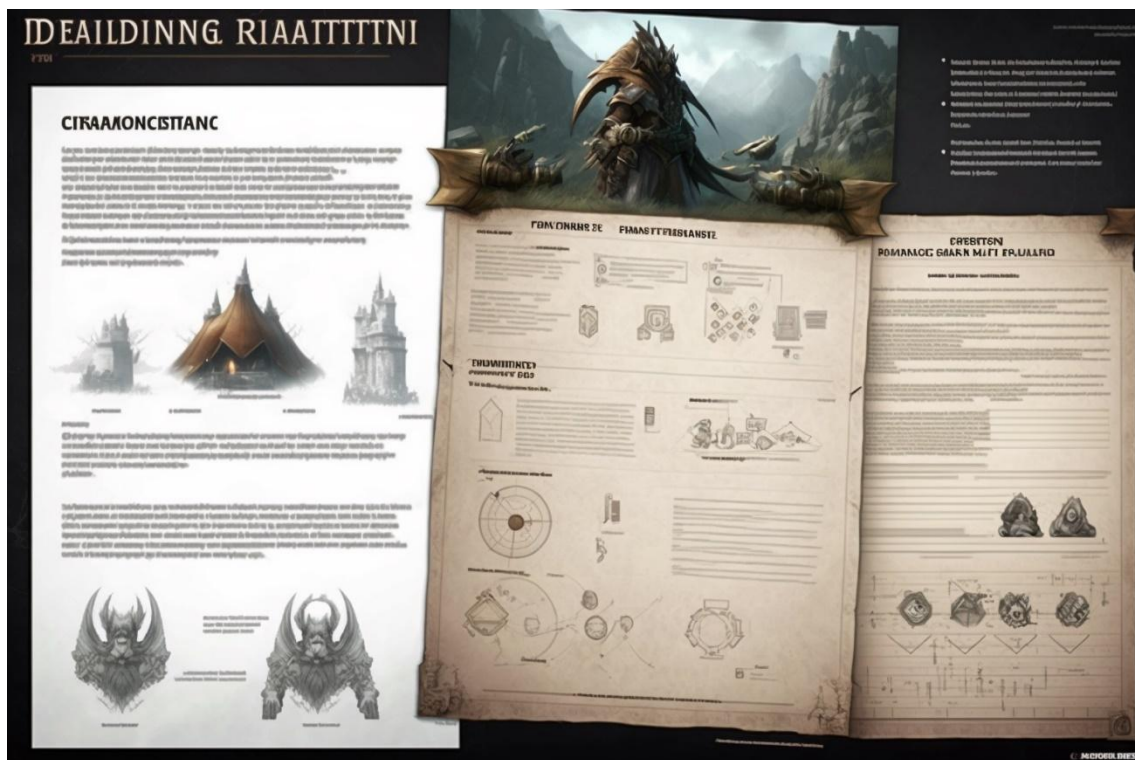
### **TEMA 3: PROPUESTA ESTÉTICA**



Una propuesta estética es un conjunto de conceptos, ideas y principios que guían la creación de una obra. En el contexto de los videojuegos, una propuesta estética se refiere al conjunto de decisiones creativas que determinan el aspecto visual y el estilo artístico del juego. Esto incluye la elección de la paleta de colores, el diseño de los personajes y entornos, el estilo de arte (por ejemplo, realista, estilizado, pixelado), la iluminación, los efectos visuales y cualquier otro elemento visual que contribuya a la experiencia estética del juego.

**Figura 1010**

*Propuesta Objetos*



Nota: La figura 10 muestra parte de su Game Design Document (GDD) donde se encuentra la estética de objetos.

Fuente: <https://nesedijital.com/a-game-design-document/>

La propuesta estética no solo busca crear un aspecto visual atractivo, sino también transmitir la atmósfera, la ambientación y la narrativa del juego, influyendo así en la inmersión del jugador en el mundo del juego y en su experiencia emocional y sensorial. Una propuesta sólida y coherente puede contribuir significativamente al éxito y la identidad única de un videojuego. Además, permite no solo definir la apariencia superficial de la obra, sino que también comunica mensajes, emociones y valores que pueden influir en la experiencia del usuario. Para definirla se puede tener en consideración los siguientes puntos:



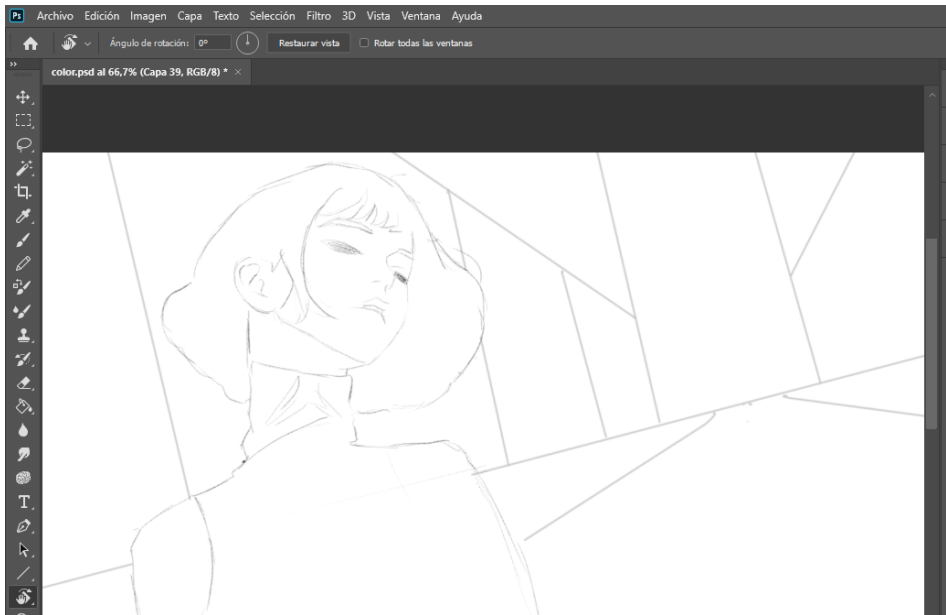
- **Investigación y Referencias:** Proceso definido por la investigación y recopilación de referencias visuales que inspiren el estilo estético deseado. Esto puede incluir obras de arte, películas, otros videojuegos, fotografías, libros u otras fuentes de inspiración relacionadas con la temática y la atmósfera del juego.
- **Definición de Temas y Conceptos:** Se identifica los temas principales y los conceptos visuales que se desee transmitir a través del juego. ¿Cuál es la historia y la ambientación del juego? ¿Qué emociones o sensaciones quieres evocar en los jugadores?
- **Paleta de Colores:** Se elijen colores que reflejen las intenciones del videojuego considerando cómo los colores pueden influir en la percepción emocional de los jugadores y en la cohesión visual del mundo del juego.
- **Estilo de Arte:** Se establece los elementos visuales específicos que caracterizan este estilo, como el nivel de detalle, la textura y la proporción. Esto debe estar enfocado siempre pensando en el target que se tiene planteado.
- **Diseño de Personajes y Entornos:** Se crea los diseños para los personajes principales, secundarios y los entornos del juego. Asegurándose que los diseños sean coherentes con la paleta de colores y el estilo de arte definidos previamente.
- **Iluminación y Ambiente:** Se establece cómo la iluminación y el ambiente pueden influir en la atmósfera del juego. Inicia la planificación sobre la iluminación natural o artificial, los efectos atmosféricos y cualquier otro elemento que contribuya a la estética visual del juego.





**Figura 1111**

*Bocetado*



Nota: En la figura 11 se puede observar el boceto de una propuesta de personaje. Elaboración: Autor.

La propuesta estética es parte fundamental de la posproducción del videojuego ya que influye en la experiencia del jugador y en la identidad visual del juego. A través de una cuidadosa planificación y ejecución, que incluye la investigación, definición de temas, selección de paleta de colores, diseño de personajes y entornos, entre otros aspectos, se puede crear un mundo visualmente coherente y atractivo que sumerja al jugador en la atmósfera y la narrativa del juego. Este proceso se debe mantener plasmado en una bitácora con los avances y referencias y de esta forma mantener un seguimiento general del desarrollo del proyecto. Se recomienda usar Photoshop para realizar los bocetos que serán modelados más tarde en programas de 3D.

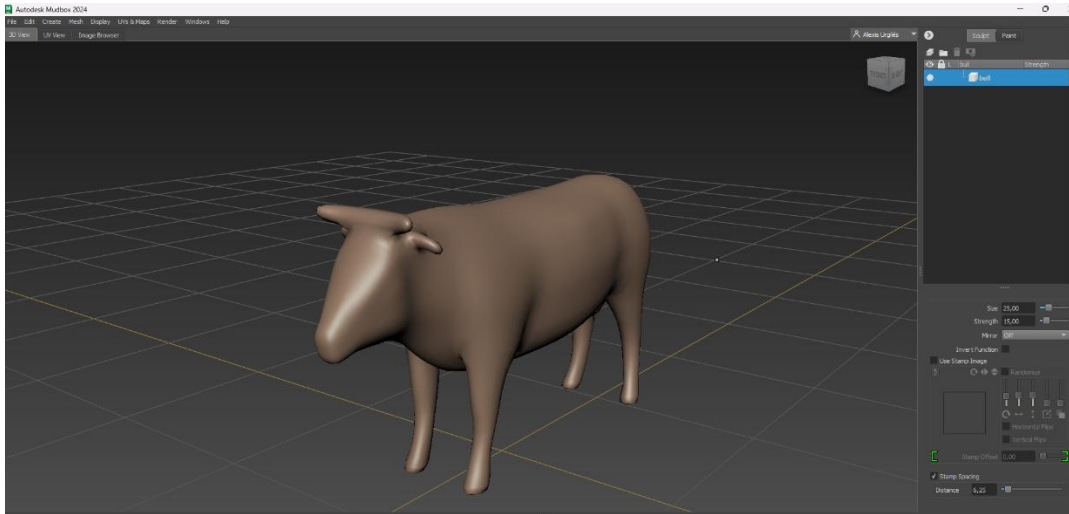
#### **TEMA 4: DISEÑO DE PERSONAJES**

Después de completar la etapa de propuesta estética se incursiona en la producción y digitalización de las ideas definidas. Es necesario tener en cuenta que el diseño de personajes en un videojuego es de los procesos más importantes en las etapas iniciales. El primer proceso que se realizará está establecido por el bocetaje del personaje con vista frontal, lateral y superior. Luego se dará lugar a la creación del personaje en 3D con sus respectivos materiales. En las etapas finales del modelado se estable los huesos y las pruebas de riggeado del personaje en funcionalidad a las acciones que sean necesarias dentro de la jugabilidad del videojuego.



**Figura 1212**

*Modelado en Mudbox*



Nota: En la figura 12 se puede observar un mesh listo para esculpir. Elaboración: Autor.

## Conceptualización

La primera etapa implica la generación de ideas y conceptos para los personajes del juego. Esto puede incluir la creación de bocetos, lluvias de ideas y la exploración de diferentes estilos y arquetipos de personajes. Se realiza este proceso en cualquier programa de edición de texto que nos permita guardar nuestra información y si existe una conceptualización gráfica se la puede hacer a mano o digitalizarlo a un programa de edición de imágenes, en este caso se estudiará Photoshop para el workflow que mantiene con los demás programas que se explorarán.

## Diseño Visual

Esta etapa del diseño del personaje establecemos definitivamente el diseño de lo que sería la estética del personaje incorporando el mensaje y objetivo que lleva el proyecto. El diseño visual implica definir la apariencia física de los personajes, incluyendo su ropa, peinado, rasgos faciales y cualquier accesorio o arma que puedan llevar. Es importante que el diseño sea coherente con el estilo artístico general del juego y que los personajes sean fácilmente reconocibles y distintivos. Antes de empezar la etapa de creación se entenderán ciertos conceptos básicos de modelado 3D.

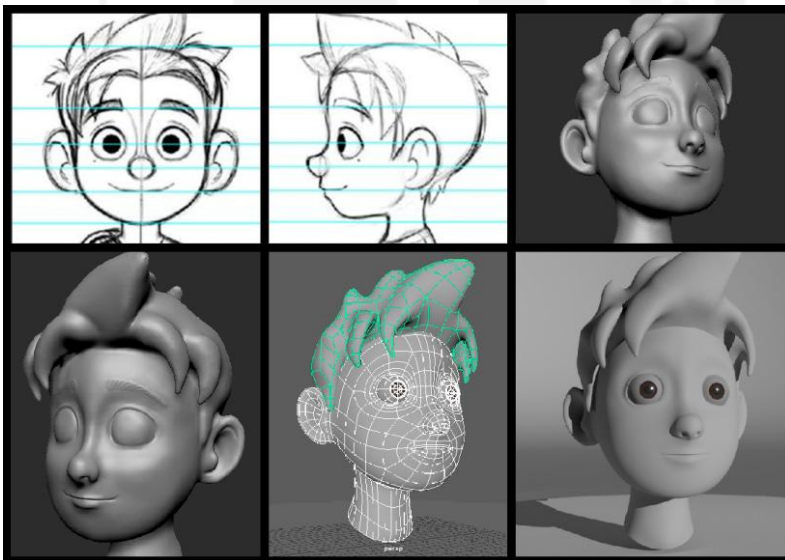
## Modelado 3D

El modelado poligonal en 3D es una técnica utilizada en el diseño de modelos tridimensionales para videojuegos y gráficos por computadora. En este método, los modelos se crean a partir de una colección de polígonos planos, como triángulos, cuadriláteros u otros polígonos de múltiples lados, que están conectados entre sí para formar una malla tridimensional. Cada vértice de la malla poligonal tiene coordenadas tridimensionales en un espacio 3D y está conectado por aristas para formar las caras de los polígonos. Estas caras definen la forma y la superficie del modelo, y su densidad y distribución determinan el nivel de detalle y suavidad del modelo.

El modelado poligonal es una de las técnicas más comunes y versátiles para crear modelos 3D en videojuegos debido a su eficiencia en términos de procesamiento y su flexibilidad para representar una amplia variedad de formas y objetos. Además, los modelos poligonales pueden ser animados fácilmente mediante la deformación de la malla, lo que los hace ideales para personajes y criaturas animadas en juegos. Sin embargo, el modelado poligonal también tiene sus limitaciones, ya que la cantidad de polígonos en el modelo puede afectar el rendimiento del juego. Por lo tanto, es importante encontrar un equilibrio entre el nivel de detalle deseado y la optimización del rendimiento para garantizar una experiencia de juego fluida y atractiva.

**Figura 1313**

*Propuesta Objetos*



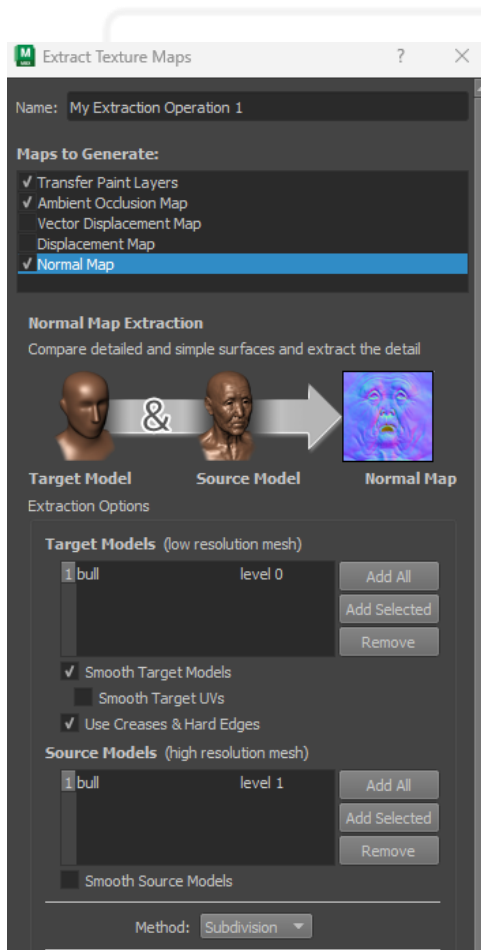
Nota: La figura 13 muestra el proceso que sigue un personaje para poder convertirlo en 3D. Fuente: <https://blogs.uninter.edu.mx/ESCAT/wp-content/uploads/2022/10/Diferencia-entre-modelado-y-animacio%CC%81n-3D.jpeg>

## Normal Map

Un mapa de normales es una textura que se utiliza para simular la apariencia de detalles de alta resolución en un modelo 3D de baja resolución. Contiene información sobre la dirección y la intensidad de las superficies rugosas, arrugadas o texturizadas, lo que permite que un modelo de baja poligonización aparezca más detallado y realista cuando se renderiza en tiempo real. Los mapas de normales se aplican al modelo 3D durante el proceso de renderizado para ajustar la forma en que la luz interactúa con la superficie del objeto, creando la ilusión de profundidad y detalle.

**Figura 1414**

*Ventana de extracción de mapas*



Nota: En la figura 14 se observa el procedimiento para extraer los mapas de los mesh diseñados. Elaboración: Autor.

## Displacement Map

Un mapa de desplazamiento es una textura que se utiliza para deformar la geometría de un modelo 3D, alterando la posición de los vértices para crear detalles y relieve. A diferencia de un

mapa de normales, que solo afecta la apariencia visual, un mapa de desplazamiento realmente modifica la forma física del modelo, empujando o tirando de los vértices en función de los valores de color de la textura. Esto permite crear detalles y características más pronunciadas y realistas en el modelo, como arrugas en la piel, rugosidades en las superficies o detalles en relieve en objetos arquitectónicos.

**Figura 1515**

*Funcionamiento de mapas*



Nota: En la figura 15 se observa el funcionamiento de las texturas y el color dentro de Mudbox. Elaboración: Autor.

## **Retopología**

Es un proceso en el modelado 3D que consiste en crear una nueva topología, o estructura de malla, para un modelo existente. Esta técnica se utiliza principalmente para optimizar la geometría de un modelo 3D, mejorando su eficiencia y calidad. Cuando se crea un modelo 3D, especialmente a través de técnicas como la escultura digital o el escaneo 3D, la malla resultante puede tener una topología irregular, con demasiados o muy pocos polígonos, bucles indeseados y geometría desordenada. Esto puede dificultar la animación, el texturizado y la manipulación del modelo, así como afectar negativamente al rendimiento en tiempo real en aplicaciones como los videojuegos.

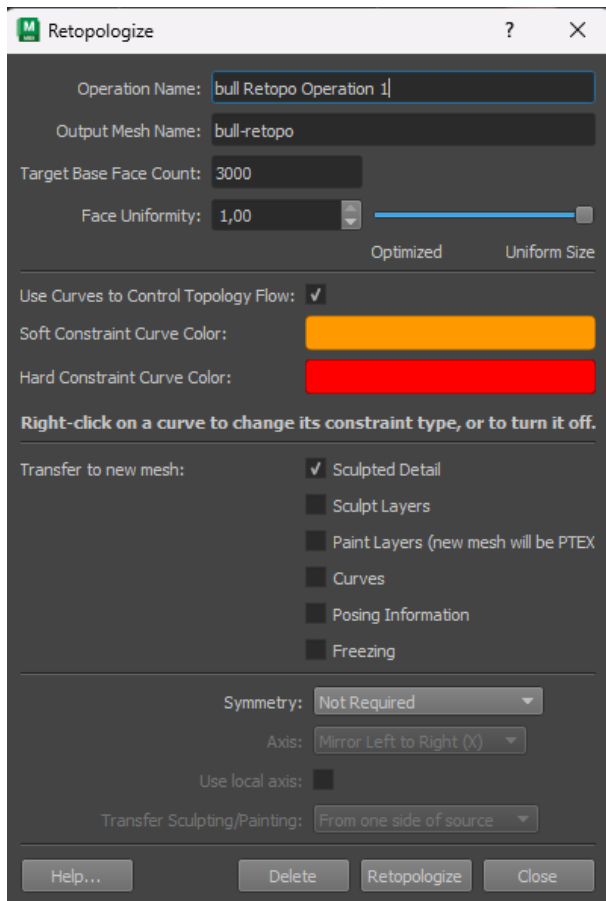
Esto implica reconstruir manualmente la geometría del modelo, utilizando técnicas como la creación de bucles de polígonos uniformes, la redistribución de la densidad de polígonos y la eliminación de detalles innecesarios.

El objetivo de la retopología es mantener la forma general y los detalles importantes del modelo original, pero con una geometría más optimizada y fácil de trabajar. Esto facilita el proceso de animación, texturizado y renderizado, así como mejora el rendimiento del modelo en tiempo

real, lo que lo hace adecuado para su uso en aplicaciones como videojuegos, realidad virtual y realidad aumentada.

**Figura 1616**

*Retopología*



Nota: En la figura 16 se observan cuáles son las limitaciones y ventajas de las retopologías en Mudbox. Elaboración: Autor.

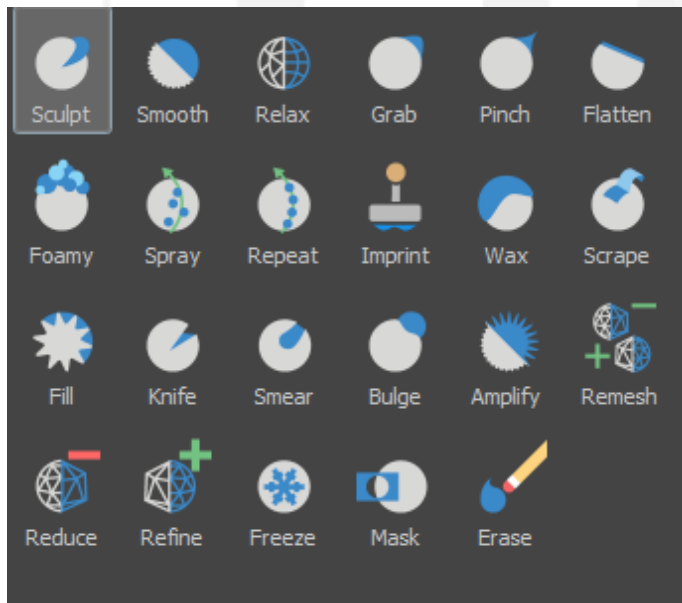
Teniendo en consideración los puntos anteriores, con el fin de crear un personaje 3D para videojuegos se pueden seguir los siguientes puntos clave.

- Bosquejo de las vistas del personaje: En Photoshop se definen las vistas que usaremos como referencia para más tarde modelar en 3D.
- Creación de la Malla Base: Se comienza creando una malla base utilizando herramientas como primitivas geométricas (cubos, esferas, cilindros) o herramientas de modelado poligonal. Esta malla servirá como estructura básica sobre la cual se construirá el modelo final. Este proceso se realizará en MAYA para poder exportarlo a MUDBOX.

- **Modelado Detallado:** Parte de la utilización de herramientas de modelado para esculpir y dar forma a la malla base para que se ajuste al diseño que se proyectó previamente. Se usará el programa MUDBOX para poder continuar con el workflow que brinda Autodesk y las herramientas que se usaran son sculpt, smooth, grab, pinch flatten, knife, budge entre otras.
- **Mapeado UV:** Realiza el mapeado UV para asignar las texturas correctamente al modelo tridimensional. Esto implica desplegar la malla en un plano 2D para que las texturas se apliquen de manera coherente y sin distorsiones. Para el corte de las UV se necesitará las herramientas de MAYA.
- **Texturización:** Se texturas y materiales al modelo para agregar color, superficies rugosas o brillantes, y otros efectos visuales. No se usarán bibliotecas externas puesto que en Mudbox se pintarán los materiales del personaje.

**Figura 1717**

*Herramientas de esculpido*

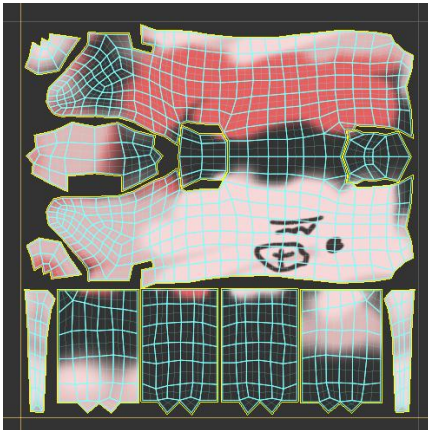


Nota: En la figura 17 se observa las herramientas disponibles para modelar el mesh. Elaboración: Autor.



**Figura 1818**

*Mapa UV*



Nota: En la figura 18 se observa cómo se ve un mapa de UVs dentro de Mudbox. Elaboración: Autor.

Es importante tener a consideración que el workflow que se está marcando está dentro de la familia de los programas de Autodesk para garantizar la veracidad de los elementos que estén siendo creados.

### **Animación**

Una vez que se ha establecido el diseño visual, se procede a crear clips que nos servirán de referencias para la ejecución de mecánicas dentro del motor de UNREAL ENGINE. Los personajes diseñados deben integrarse en el juego de manera efectiva, lo que implica, la creación de interacciones con otros personajes y elementos del entorno, y asegurarse de que su comportamiento y apariencia sean coherentes en todas las situaciones.

La creación del personaje está constituida por una serie de etapas que pueden ser modificadas dependiendo del estilo de trabajo del diseñador y modelador, la etapa de boceteo puede ser evitada y puede mantener un diseño con acabados simples o puede ser necesario una ilustración con fines de Splash Art que sirve como base para la creación del personaje en 3D con acabados muy elaborados.

## **TEMA 5: ASSETS**

En el diseño de videojuegos, un "asset" se refiere a cualquier elemento digital utilizado en la creación del juego, como gráficos, modelos 3D, animaciones, música, efectos de sonido, scripts de programación, archivos de texto, y más. Los assets pueden ser creados por el equipo de desarrollo del juego o adquiridos de bibliotecas de activos o de terceros. Los assets son componentes fundamentales para la construcción de un videojuego y pueden ser utilizados para crear los personajes, entornos, objetos, efectos visuales y sonoros, interfaces de usuario, y cualquier otro elemento presente en el juego. La calidad y variedad de los assets pueden influir en la apariencia, jugabilidad y experiencia general del juego.

**Figura 1919**

*Herramientas Mudbox*



Nota: En la figura 11 se observa las herramientas disponibles que permiten la creación de texturas para los objetos.  
Elaboración: Autor.

El término "asset" también puede referirse a activos digitales utilizados para otros propósitos en el diseño de videojuegos, como recursos de marketing, material promocional, documentos de diseño, o cualquier otro elemento necesario para el desarrollo, distribución y promoción del juego. Los assets son los bloques de construcción esenciales que componen un videojuego y se dividirá la creación de cada uno de ellos dependiendo de la importancia para el proyecto.

### **Objetos**

La creación de objetos que irán incorporados dentro del programa serán realizados con un proceso similar a la creación de personajes, con la diferencia que el bocetaje principal no es necesario dependiendo la importancia del objeto. Es decir, cuando se diseñan piedras no llegan

a tener el mismo impacto que el diseño de la casa del personaje principal, pero ambos pueden llegar a tener una etapa de modelado similar.

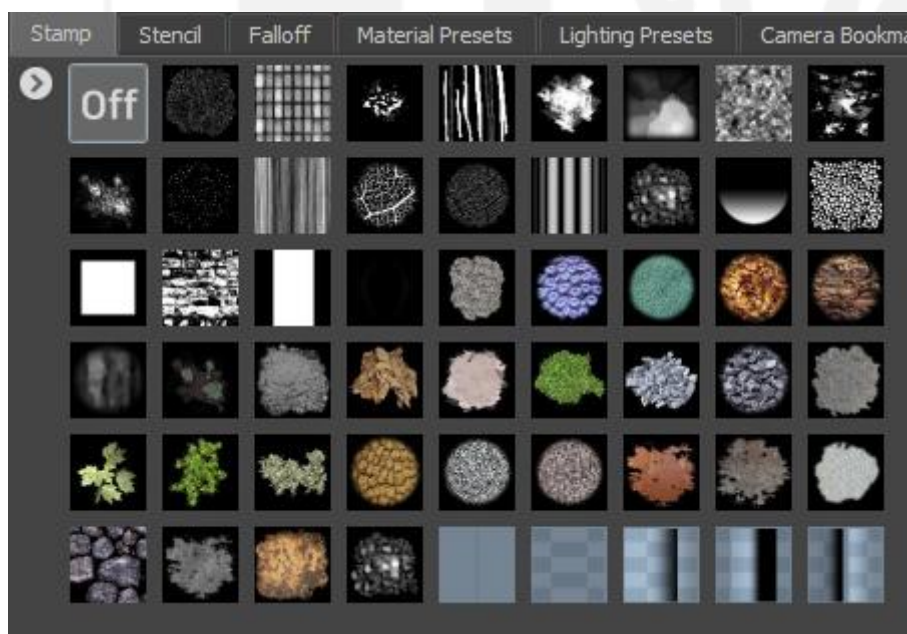
### Escenario

El diseño de escenarios y terrenos para videojuegos es un proceso fundamental que implica la creación de entornos virtuales que los jugadores puedan explorar. El terreno se modelará utilizando MAYA y las herramientas de diseño d Unreal Engine puesto que permite dar forma a la topografía del entorno. Esto incluye la creación de montañas, colinas, valles, ríos, lagos y cualquier otro elemento geográfico que sea necesario para el entorno del juego.

Una vez que se ha modelado el terreno, se aplican texturas para darle color y detalle. Esto puede incluir texturas para la vegetación, la tierra, la roca, el agua y otros elementos del entorno. Se utilizan técnicas de mapeado UV para asignar las texturas al terreno de manera coherente y realista. Se añaden elementos decorativos y detalles al escenario, como árboles, arbustos, rocas, edificios, caminos, señales, vegetación, etc. Estos elementos ayudan a enriquecer el entorno y a crear una experiencia más inmersiva para el jugador.

**Figura 2020**

*Herramientas textura*



Nota: En la figura 12 se observa las herramientas que permiten la creación de relieve y texturas. Elaboración: Autor.

Al final se configura la iluminación del escenario utilizando fuentes de luz naturales y artificiales para crear una atmósfera adecuada y realista. Esto incluye la dirección, intensidad, color y sombras de la luz para mejorar la apariencia visual del entorno. Es necesario recordar que el escenario está ligado directamente a la jugabilidad del proyecto para poder realizar la programación de mecánicas en relación al terreno y los objetos que la componen.

### **Diorama**

Se plantea la realización de un diorama para la exposición del proyecto dentro de un teaser que se presentará al final de la unidad y para esto es necesario entender todos los conceptos que lo envuelven. Un diorama es una representación tridimensional de una escena, evento o paisaje, típicamente contenida dentro de un espacio limitado, como una caja o un marco. Estas representaciones, elaboradas con una variedad de materiales como cartón, papel o arcilla, capturan detalles minuciosos y realistas, y se utilizan en diversos contextos, desde proyectos educativos hasta maquetas arquitectónicas y recreaciones históricas. En el diseño de videojuegos, los dioramas son útiles para visualizar y planificar los escenarios y niveles, proporcionando una representación tangible del entorno del juego antes de su implementación final. Para su realización se plantea un diseño de unificación de todos los assets previamente realizados con un enfoque al desarrollo próximo de lo que será las mecánicas del personaje principal en el terreno.

### **Autoevaluación 2**

1. ¿Cuáles son los elementos clave a considerar al diseñar un personaje para un videojuego?
2. ¿Qué importancia tiene la coherencia visual del diseño del personaje con la narrativa del juego?
3. ¿Cómo afecta el estilo artístico del juego al diseño del personaje?
4. ¿Qué herramientas y software se utilizan comúnmente en el diseño de personajes para videojuegos?
5. ¿Puedo realizar retopología en Maya y Mudbox?
6. ¿Qué elementos debe incluir una propuesta de arte para un videojuego?





7. ¿Cuál es la importancia de establecer una paleta de colores y un estilo visual coherente en la propuesta de arte?
8. ¿Cómo se pueden comunicar las ideas y conceptos artísticos de manera efectiva en una propuesta de arte?
9. ¿Qué consideraciones se deben tener en cuenta al presentar una propuesta de arte a un equipo de desarrollo de videojuegos?
10. ¿Cómo se integra la retroalimentación del equipo y de los interesados en la propuesta de arte para mejorar y refinar el diseño final?

## Resumen de la Unidad 2

El diseño de videojuegos abarca una serie de temas fundamentales, el diseñador se encarga de moldear y ejecutar el proyecto desde la etapa de posproducción hasta la producción, trabajando en la conceptualización de la idea principal, la propuesta estética y el diseño de personajes y escenarios. Además, se explora la narrativa en los videojuegos, donde el diseñador debe entender cómo la historia guiará las bases del proyecto, teniendo en cuenta elementos de acción, diseño y mecánicas de juego. La propuesta estética, por su parte, busca transmitir la atmósfera, la ambientación y la narrativa del juego, influyendo en la inmersión del jugador. Se detallan procesos como el diseño de personajes, la creación de assets y la configuración del escenario, con énfasis en la importancia de la coherencia visual y la optimización del rendimiento. Finalmente, se plantea la realización de un diorama como una representación tridimensional de la escena del juego, que sirve para visualizar y planificar los escenarios antes de su implementación final.





## UNIDAD 3 RIGG Y SUS CONTROLADORES

### Temas y Subtemas

#### Tema 1: Rigging

#### Tema 2: Controladores

#### Tema 3: Pesos

### TEMA 1: RIGGING

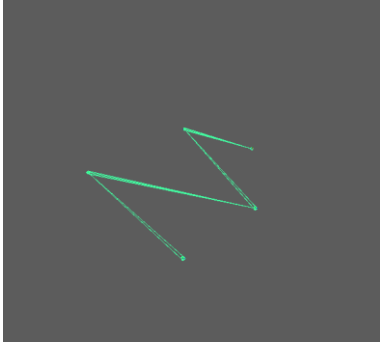
El rigging es un proceso en el campo de la animación 3D que implica la creación de una estructura virtual, llamada esqueleto o armadura, dentro de un modelo tridimensional. Este esqueleto se compone de huesos virtuales, articulaciones y controles que permiten animar el modelo de manera fluida y realista.





**Figura 2121**

*Rig*



Nota: En la figura 21 se observa cómo se ve un rigg básico. Elaboración: Autor.

El objetivo del rigging es dotar al modelo de la capacidad de movimiento, de manera que pueda moverse de forma natural y expresiva, como lo haría un personaje o criatura real. Los artistas de rigging configuran y ajustan la posición, orientación y comportamiento de los huesos y articulaciones para lograr los efectos de movimiento deseados.

**Figura 2222**

*Herramientas de Rig*

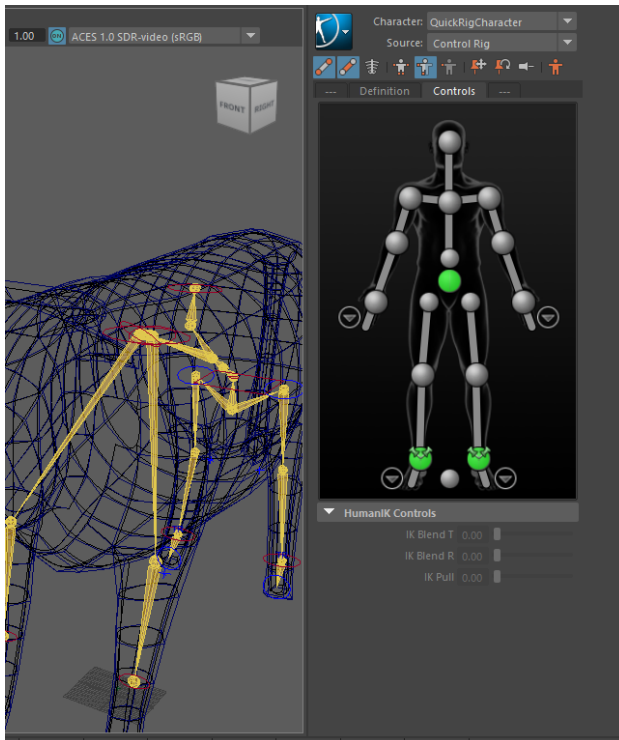


Nota: En la figura 22 se observa las herramientas destinadas al riggeado. Elaboración: Autor.

Además de los huesos y las articulaciones, el rigging también puede involucrar la aplicación de otros elementos como controladores, deformadores y restricciones, que permiten a los animadores manipular y controlar el movimiento del modelo con mayor precisión y eficiencia.

**Figura 2323**

*Herramientas*



Nota: En la figura 23 se observa la estructura del riggeado en Maya. Elaboración: Autor.

## TEMA 2: CONTROLADORES

Los controladores de rigging son objetos que se utilizan para manipular y controlar el movimiento y la deformación de los modelos tridimensionales durante la animación. Estos controladores permiten a los animadores realizar ajustes precisos y realizar animaciones complejas de manera más eficiente. Se asociará los pasos para su creación de la siguiente manera.

### Creación de Controladores

Los controladores de rigging se crean como objetos específicos, como esferas, cubos u otros objetos geométricos simples que se colocan en la escena y asociados a las partes relevantes del esqueleto o estructura de rigging del modelo.

### Asociación con los huesos

Los controladores se vinculan a los huesos o articulaciones del esqueleto del modelo mediante restricciones de control (constraints) o mediante conexiones directas. Esto permite que los

controladores sigan los movimientos y transformaciones de los huesos, lo que a su vez afecta el comportamiento del modelo.

### Deformación

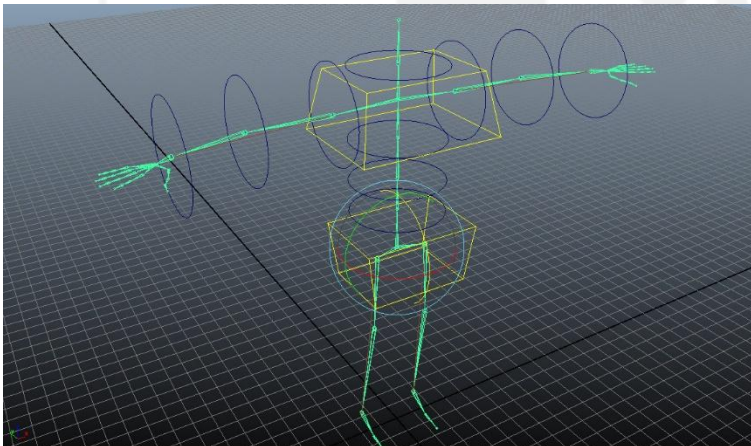
Además de controlar el movimiento del modelo, los controladores de rigging también pueden utilizarse para controlar la deformación de la malla del modelo, como la flexión de los músculos o la expresión facial. Esto se logra mediante la aplicación de controladores adicionales, como deformadores y blendshapes, que están vinculados a los controladores principales.

### Jerarquía

Los controladores de rigging se organizan en una jerarquía lógica que refleja la estructura del modelo y sus relaciones de control. Esto facilita la manipulación y animación del modelo, ya que los animadores pueden trabajar con controladores de nivel superior para afectar al modelo en su totalidad, o con controladores individuales para ajustes más detallados.

#### Figura 2424

Controladores



Nota: La figura 24 muestra la visibilidad de los controladores dentro de Maya. Fuente: <https://i.snipboard.io/zfYmd.jpg>

## TEMA 3: PESOS

Una de las tareas más importantes que tienen los encargados del riggeado en un objeto poligonal es la definición correcta de pesos para el riggeado puesto que esto define la movilidad natural y fluida de nuestros personajes. En rigging, los "pesos" se refieren a la influencia que

tienen ciertos componentes de un esqueleto (como huesos, articulaciones o controladores) sobre la deformación de la malla de un modelo tridimensional. Estos pesos determinan cómo se mueve y se deforma la malla del modelo cuando se realiza una animación.

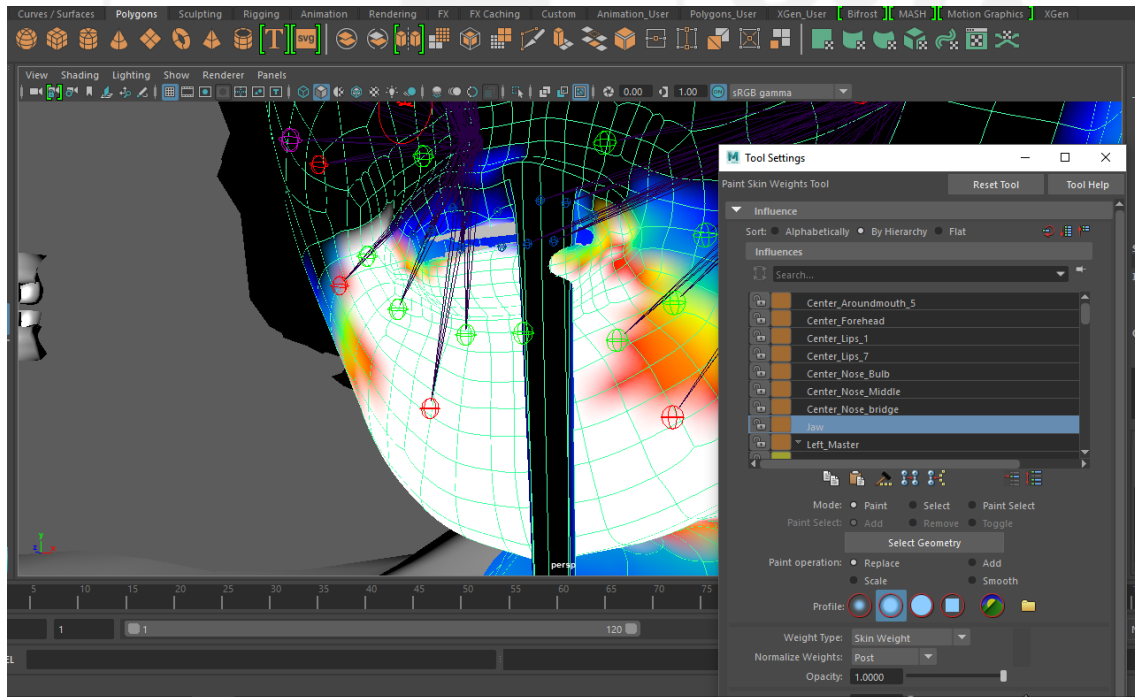
Cada vértice de la malla de un modelo 3D puede estar asociado con uno o más huesos del esqueleto, y los pesos indican cuánta influencia tiene cada hueso en la deformación de ese vértice en particular. Por ejemplo, en un personaje humano, los huesos de la pierna tendrían más peso en los vértices de la pierna que en los de los brazos.

Los pesos se representan como valores numéricos que van de 0 a 1, donde 0 significa que el hueso no tiene influencia en absoluto sobre el vértice, y 1 significa que el hueso tiene la máxima influencia. Los valores entre 0 y 1 indican una influencia parcial del hueso en el vértice.

El ajuste de los pesos es una parte crucial del proceso de rigging, ya que determina cómo se distribuye el movimiento entre los diferentes componentes del esqueleto y cómo se comporta la malla del modelo durante la animación. Los artistas de rigging ajustan los pesos cuidadosamente para lograr animaciones fluidas y realistas, asegurándose de que los movimientos del modelo se vean naturales y coherentes.

**Figura 2525**

*WeightPainting*





Nota: La figura 25 muestra el proceso erróneo de pintura de pesos. Fuente: <https://polycount.com/discussion/190192/maya-weight-painting-issue-vertices-getting-pulled-towards-ground>

### Autoevaluación 3

1. ¿Qué es el rigging en animación 3D y cuál es su propósito en la creación de modelos tridimensionales?
2. ¿Cuáles son los componentes principales de un rig en animación 3D y cómo interactúan entre sí para permitir el movimiento del modelo?
3. ¿Por qué es importante el rigging en la animación de personajes y objetos en un entorno 3D?
4. ¿Cuáles son las diferencias entre un rig de personaje y un rig de objeto inanimado en términos de complejidad y funcionalidad?
5. ¿Qué papel juegan los deformadores y las restricciones en el proceso de rigging y cómo afectan al movimiento del modelo?
6. ¿Qué son los controladores en animación 3D y cuál es su función principal en el proceso de animación?
7. ¿Cuáles son los diferentes tipos de controladores que se pueden utilizar en la animación de personajes y objetos en 3D?
8. ¿Cómo se crean y se asignan controladores a partes específicas de un rig para manipular el movimiento y la deformación durante la animación?
9. ¿Qué ventajas ofrece el uso de controladores en términos de eficiencia y precisión en la animación 3D?
10. ¿Cómo se organiza y se jerarquiza la estructura de controladores en un rig para facilitar la manipulación y la animación fluida del modelo?





### Resumen de la Unidad 3

El rigging en animación 3D es un proceso esencial que implica la construcción de una estructura virtual dentro de un modelo tridimensional, conocida como esqueleto o armadura. Esta estructura está compuesta por huesos virtuales, articulaciones y controles que permiten animar el modelo con fluidez y realismo, otorgándole una capacidad de movimiento natural y expresivo similar al de un personaje real. Además de los elementos básicos como huesos y articulaciones, el rigging puede involucrar el uso de controladores, deformadores y restricciones para lograr un control preciso y eficiente del movimiento.

## UNIDAD 4 PROGRAMACIÓN EN UNREAL ENGINE

### Temas y Subtemas

**Tema 1: Introducción a la plataforma**

**Tema 2: Level Blueprint y Blueprint Class**

**Tema 3: Programación del Personaje**

**Tema 4: Diseño de Interfaz de Usuario**

**Tema 5: Eventos**

### TEMA 1: INTRODUCCIÓN DE LA PLATAFORMA

UE será el motor de trabajo dentro de este proyecto y se utilizará las herramientas que ofrece la sección de videojuegos en ThirdPersonCharacter. Se explorará la definición de Unreal Engine



y el funcionamiento de su plataforma. UE es un motor de juego líder en la industria de los videojuegos, utilizado por desarrolladores de todo el mundo para crear experiencias interactivas de alta calidad. Con su potente capacidad de renderizado en tiempo real, herramientas intuitivas de desarrollo de contenido y soporte multiplataforma, Unreal Engine permite a los creadores materializar sus visiones creativas y llevar a los jugadores a mundos virtuales sorprendentes y envolventes. Desde juegos independientes hasta grandes producciones AAA, Unreal Engine se ha convertido en un pilar fundamental en la creación de algunos de los títulos más exitosos y visualmente impactantes del mercado de los videojuegos.

**Figura 2626**

*Pantalla de carga*



Nota: En la figura 26 se observa el logo y una muestra de las posibilidades de diseño de espacios dentro de UE5.  
Elaboración: Autor.

## TEMA 2: LEVEL BLUEPRINT Y BLUEPRINT CLASS

### Blueprints

Esta unidad explorará la programación por blueprints y se entenderá su uso en la programación dentro de Unreal Engine puesto que será la raíz del proyecto de videojuego que se está realizando. Los Blueprints en Unreal Engine son un sistema de programación visual que permite a los desarrolladores crear lógica de juego y comportamientos interactivos sin necesidad de escribir código tradicional. En lugar de escribir líneas de código, los desarrolladores utilizan nodos y conexiones para crear secuencias de acciones y decisiones que determinan cómo se



comporta el juego en respuesta a diferentes eventos, acciones del jugador o condiciones del entorno.

Los Blueprints están diseñados para ser intuitivos y accesibles, lo que permite a una amplia gama de personas, incluyendo artistas y diseñadores, crear contenido interactivo y funcional para sus juegos. Con el sistema de Blueprints, los desarrolladores pueden prototipar ideas rápidamente, ajustar la jugabilidad en tiempo real y crear mecánicas de juego complejas sin necesidad de tener un profundo conocimiento de programación.

Los Blueprints pueden utilizarse para una variedad de propósitos en Unreal Engine, incluyendo la creación de inteligencia artificial para personajes no jugadores, la implementación de sistemas de inventario y progresión, la creación de interfaces de usuario interactivas, la configuración de eventos de juego, la animación de objetos y personajes, entre otros. En resumen, los Blueprints son una herramienta poderosa y versátil que permite a los desarrolladores dar vida a sus ideas y crear experiencias interactivas de alta calidad en Unreal Engine sin la necesidad de programación tradicional.

### **Blueprint Class**

Los Blueprint Class pueden representar una amplia variedad de elementos en el juego, como personajes jugables, enemigos, objetos, armas, vehículos, y más. Cada Blueprint Class contiene una colección de nodos y eventos que definen cómo se comporta y responde el objeto en el juego. Una de las ventajas clave de utilizar Blueprint Classes es su capacidad para iterar rápidamente y ajustar el comportamiento de los objetos sin necesidad de modificar directamente el código fuente del juego. Esto permite a los diseñadores y artistas experimentar con diferentes mecánicas de juego y prototipos de manera rápida y eficiente.

Además, los Blueprint Classes admiten la herencia, lo que significa que una clase puede heredar propiedades y funcionalidades de otra clase. Esto facilita la creación de objetos similares que comparten ciertas características y comportamientos, lo que ahorra tiempo y reduce la necesidad de crear nuevos Blueprints desde cero.

### **Level Blueprint**

El Level Blueprint es un tipo especial de Blueprint en Unreal Engine que se utiliza para controlar la lógica y los eventos específicos que ocurren dentro de un nivel o escena del juego. A diferencia de los Blueprint Class, que se utilizan para definir el comportamiento de objetos individuales en



el juego, el Level Blueprint está asociado con un nivel en particular y puede controlar eventos que ocurren en ese nivel en tiempo real.

El Level Blueprint proporciona una manera poderosa de crear interacciones y eventos personalizados dentro de un nivel. Los desarrolladores pueden utilizar nodos y conexiones para crear secuencias de acciones y reacciones que se activan en respuesta a diferentes eventos, acciones del jugador o condiciones del entorno.

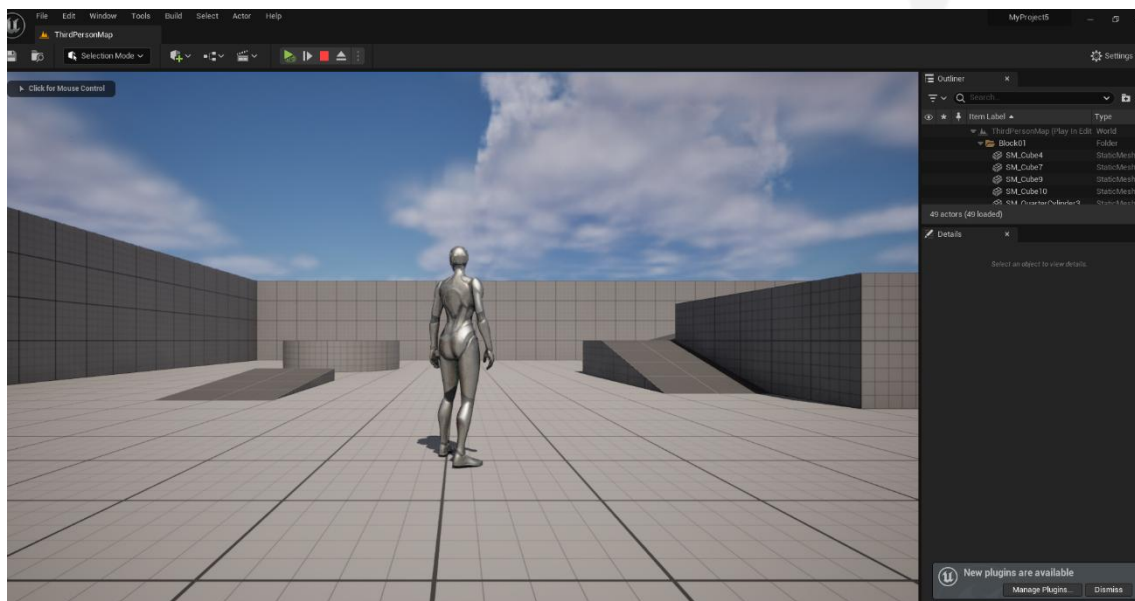
Por ejemplo, el Level Blueprint se puede utilizar para activar trampas, cambiar la iluminación, reproducir animaciones, iniciar secuencias de eventos y mucho más. Se puede trabajar directamente dentro del editor de Unreal Engine para asignar acciones y comportamientos a elementos específicos del nivel, como objetos, personajes, activadores de eventos y zonas del mapa. El Level Blueprint es especialmente útil para prototipar ideas rápidamente, ajustar la jugabilidad en tiempo real y crear mecánicas de juego complejas que afectan a todo el nivel. También es una herramienta poderosa para crear eventos de juego específicos de la historia, desencadenar cinemáticas y gestionar la progresión del jugador dentro del juego.

### TEMA 3: PROGRAMACIÓN DEL PERSONAJE

Realizar la programación del personaje con sus funciones principales como caminar, correr, saltar, disparar es sencillo en UE. Existe Blueprints ya establecidos por default que nos sirven de guía para su creación o cambio de ajustes.

**Figura 2727**

*Funcionamiento de mapas*



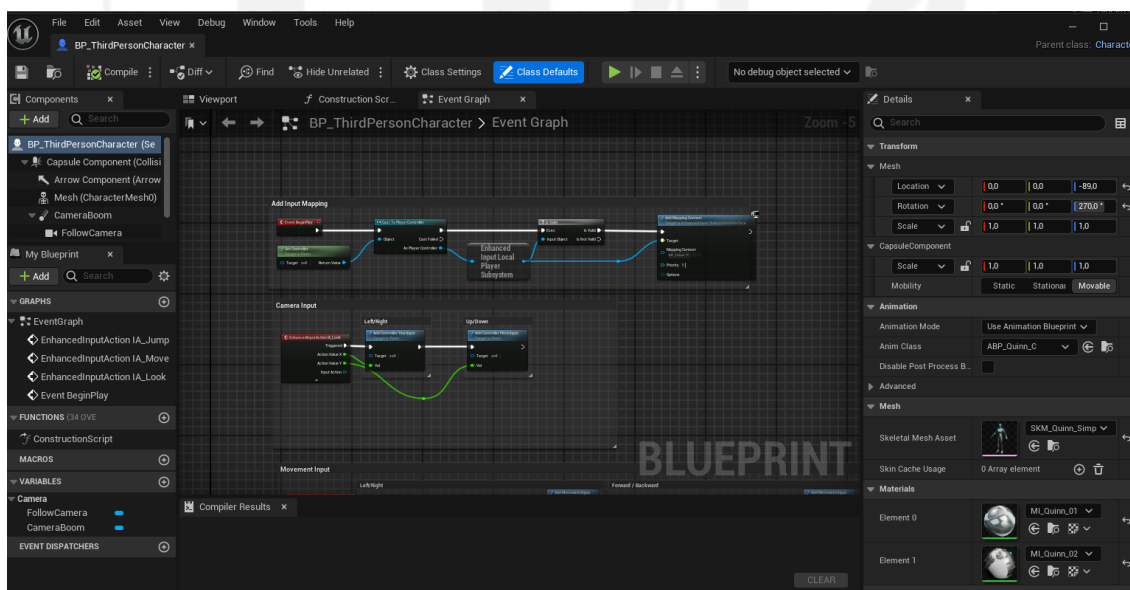
Nota: En la figura 27 se observa cómo se presenta el personaje de tercera persona dentro de la simulación de UnrealEngine. Elaboración: Autor.

También es posible añadir más funciones al personaje con programación de nuevos blueprints dependiendo de las acciones o eventos que se deseen tener. La perspectiva del personaje principal está definida al inicio de la creación del personaje. En este proyecto se establece un personaje en tercera persona. Para realizar sus ajustes se debe tener en consideración:

- Configuración de los componentes como cámaras, mallas, colisionadores.
- Configuración de la malla del personaje, las animaciones y otros aspectos visuales del personaje.
- Programación de nodos con eventos y acciones para controlar el movimiento, la interacción y otros comportamientos del personaje.
- Ajuste de acciones principales y nodos de colisión para detectar y responder a colisiones con el entorno y otros objetos.

**Figura 2828**

*Programación del ThirdpersonCharacter*



Nota: En la figura 28 se observa la programación base que presenta ThirdPersonCharacter. Elaboración: Autor.



## TEMA 4: INTERFAZ DE USUARIO

En esta etapa de diseñar y se aplicara el UI la cual debe mantener la lógica del etilo artístico que se ha logrado a lo largo del proyecto y la narrativa que ha mantenido el videojuego. Por esta razón se entenderá primero conceptos básicos sobre esta sección. La UI, o Interfaz de Usuario, se refiere a todos los elementos visuales y táctiles que permiten a los usuarios interactuar con un sistema informático, dispositivo electrónico, aplicación o software. La UI incluye todo lo que el usuario ve y con lo que puede interactuar en la pantalla, como botones, menús, barras de desplazamiento, campos de texto, iconos, indicadores, entre otros.

El propósito de la UI es facilitar la comunicación entre el usuario y el sistema, proporcionando una experiencia de usuario intuitiva, eficiente y agradable. Una buena interfaz de usuario debe ser fácil de entender y utilizar, permitiendo al usuario realizar acciones y acceder a funciones de manera rápida y sin esfuerzo.

La UI puede variar ampliamente según el tipo de dispositivo o sistema en el que se utilice. Por ejemplo, la UI de un sistema operativo para computadoras puede incluir elementos como el escritorio, el menú de inicio, la barra de tareas y las ventanas de las aplicaciones y en los juegos se aplica en los botones de menú de inicio de pause de victoria y derrota.

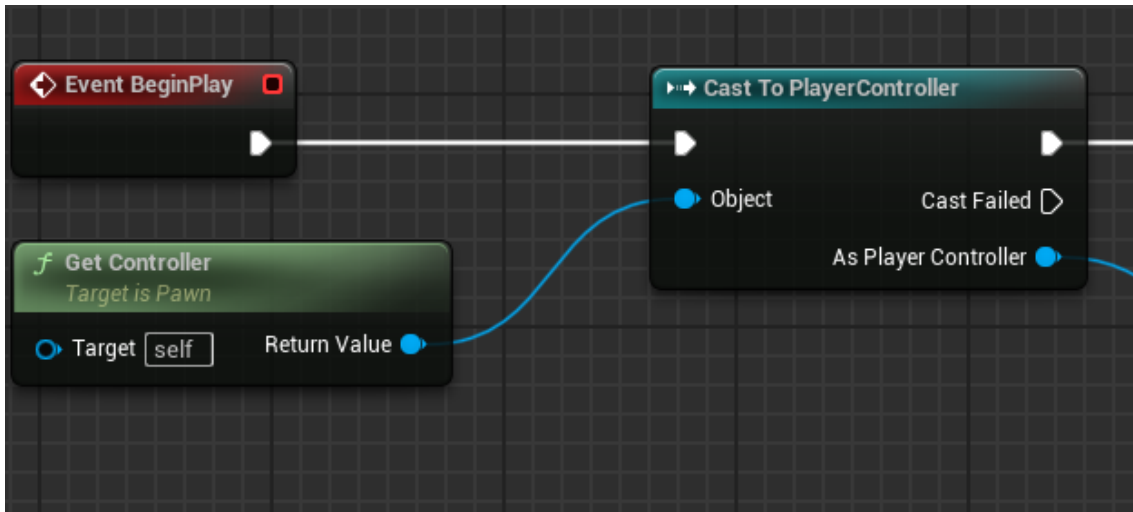
## TEMA 5: EVENTOS

En términos de desarrollo de videojuegos en Unreal Engine, los eventos son acciones que desencadenan respuestas o comportamientos dentro del juego. Estos eventos pueden ser de diversos tipos y pueden incluir acciones como presionar un botón, mover un joystick, colisiones entre objetos, temporizadores que han expirado, señales de entrada del usuario, cambios en el estado del juego, entre otros.



**Figura 2929**

*Event BeginPlay*

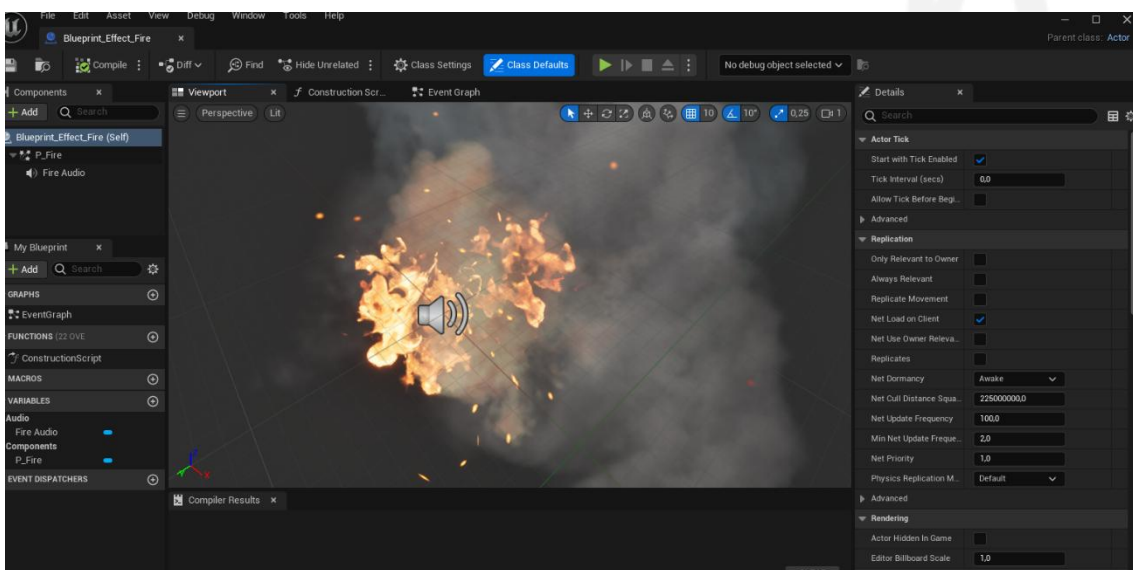


Nota: En la figura 29 se observa la programación de un evento dentro de los blueprints. Elaboración: Autor.

Los eventos son fundamentales en la programación de videojuegos porque permiten que el juego reaccione y responda dinámicamente a las acciones del jugador y a otros eventos que ocurran durante la ejecución del juego. Por ejemplo, un evento de colisión entre el jugador y un objeto puede desencadenar una acción de daño al jugador, o un evento de temporizador puede activar una animación de explosión después de cierto tiempo.

**Figura 3030**

*Trigger*



Nota: En la figura 30 se observa un efecto de fuego que puede ser añadido con una colisión. Elaboración: Autor.

La materia está designada para el diseño del videojuego por esta razón no se profundizará en la programación del UI, pero si se enfatizara en el diseño que este tendrá. Para esto se podrá realizarlo dentro de Illustrator o Photoshop.

#### Autoevaluación 4

1. ¿Cuál es la función principal de los Blueprints en Unreal Engine?
2. ¿Qué ventajas ofrecen los Blueprints en comparación con la programación tradicional?
3. ¿Qué elementos pueden representar los Blueprint Class en un videojuego?
4. ¿Cuál es la diferencia entre Blueprint Class y Level Blueprint en Unreal Engine?
5. ¿Cómo se utilizan los controladores de rigging en la animación 3D?
6. ¿Qué son los pesos en el rigging y por qué son importantes en la animación?
7. ¿Cuál es el propósito de la interfaz de usuario (UI) en un videojuego?
8. ¿Por qué son fundamentales los eventos en el desarrollo de videojuegos en Unreal Engine?
9. ¿Cuál es el papel del Level Blueprint en la creación de interacciones dentro de un nivel de juego?
10. ¿Cómo se pueden agregar nuevas funciones al personaje en un juego utilizando Blueprints?

#### Resumen de la Unidad 4

En la unidad se presenta la plataforma de UE5 y se comprende que los blueprints tienen el poder de simplificar muchos pasos, pero se debe tener un conocimiento profundo en lógica y estructuración de elementos para poder manejar adecuadamente todas las herramientas que UE entrega. Además, imprescindible tener en cuenta que los Blueprints ofrecen una forma intuitiva y poderosa de crear experiencias interactivas en las cuales se incluye la creación del UI y programación de eventos del personaje y objetos.



## UNIDAD 5 TESTING-TEASER

### Temas y Subtemas

#### Tema 1: Empaquetado

#### Tema 2: Testing

#### Tema 3: Teaser

### TEMA 1: EMPAQUETADO

El empaquetado consiste en el proceso de compilar y crear una versión ejecutable del videojuego, esta se encuentra lista para ser distribuida y ejecutada en diferentes plataformas. Este proceso prepara todos los archivos y recursos necesarios del juego, incluyendo activos gráficos, código de programación, sonidos, configuraciones y otros elementos, para que el juego pueda ser instalado y ejecutado en dispositivos específicos, como computadoras, consolas de videojuegos o dispositivos móviles. El proceso de empaquetado en Unreal Engine varía dependiendo de la plataforma de destino del juego, ya que cada una tiene sus propias especificaciones y requisitos técnicos. Por ejemplo, el empaquetado para PC puede requerir diferentes configuraciones y ajustes que el empaquetado para consolas de videojuegos o dispositivos móviles.

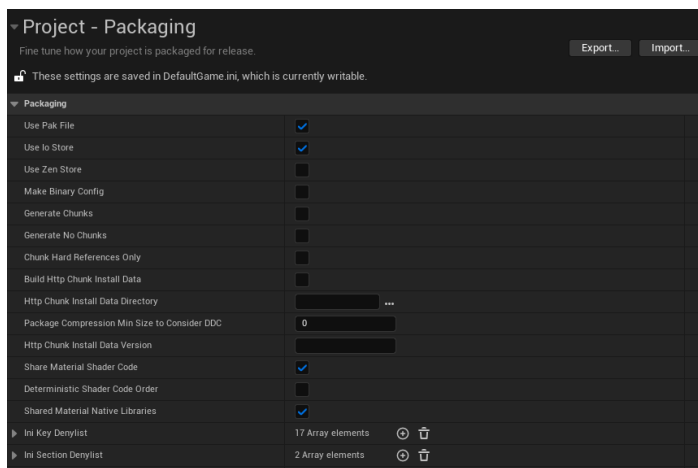




Se debe tener en consideración la configuración general del proyecto y las variantes de sus ajustes dependiendo la plataforma de destino, el rendimiento, resolución de pantalla, controles de entrada, entre otros. La compilación del código de programación debe ser revisada y ajustada en caso de tener errores, esto permite empaquetar el videojuego en un proyecto ejecutable.

**Figura 3131**

### Packaging



Nota: En la figura 31 se observa la pantalla donde se realizará los cambios de empaquetado. Elaboración: Autor.

## TEMA 2: TESTING

Testing es un proceso fundamental en el desarrollo de videojuegos que consiste en verificar y validar que un sistema o una aplicación cumpla con los requisitos y expectativas establecidos. El objetivo principal del testing es identificar y corregir errores, defectos o fallos en el software antes de su lanzamiento al público, garantizando así su calidad, fiabilidad y funcionamiento adecuado.

El proceso de testing implica diseñar y ejecutar una serie de pruebas específicas para evaluar diferentes aspectos del software, como su funcionalidad, rendimiento, seguridad, usabilidad y compatibilidad con diferentes dispositivos o sistemas operativos. Estas pruebas pueden realizarse de forma manual, mediante la intervención humana, o de forma automatizada, utilizando herramientas y scripts diseñados para ejecutar pruebas repetitivas y exhaustivas.

## TEMA 3: TEASER

El objetivo final de la materia es realizar un teaser que demuestre lo aprendido dentro de todas las 5 unidades digitalizando conceptos, personajes, escenarios y mecánicas. Un teaser de un



videojuego es un breve avance o adelanto promocional diseñado para generar anticipación y expectativa entre los jugadores antes del lanzamiento oficial del juego. Por lo general, los teasers de videojuegos presentan secuencias cinemáticas o fragmentos de jugabilidad que ofrecen una visión intrigante pero limitada del mundo, la historia o las mecánicas del juego.

### **Secuencer**

En Unreal Engine, Sequencer es una herramienta de creación de cinemáticas y animaciones utilizada para producir contenido visual de alta calidad dentro del motor de juego. Secuencer permite a los desarrolladores y artistas crear secuencias de cinemáticas, escenas y animaciones de manera intuitiva y eficiente, sin necesidad de utilizar software externo.

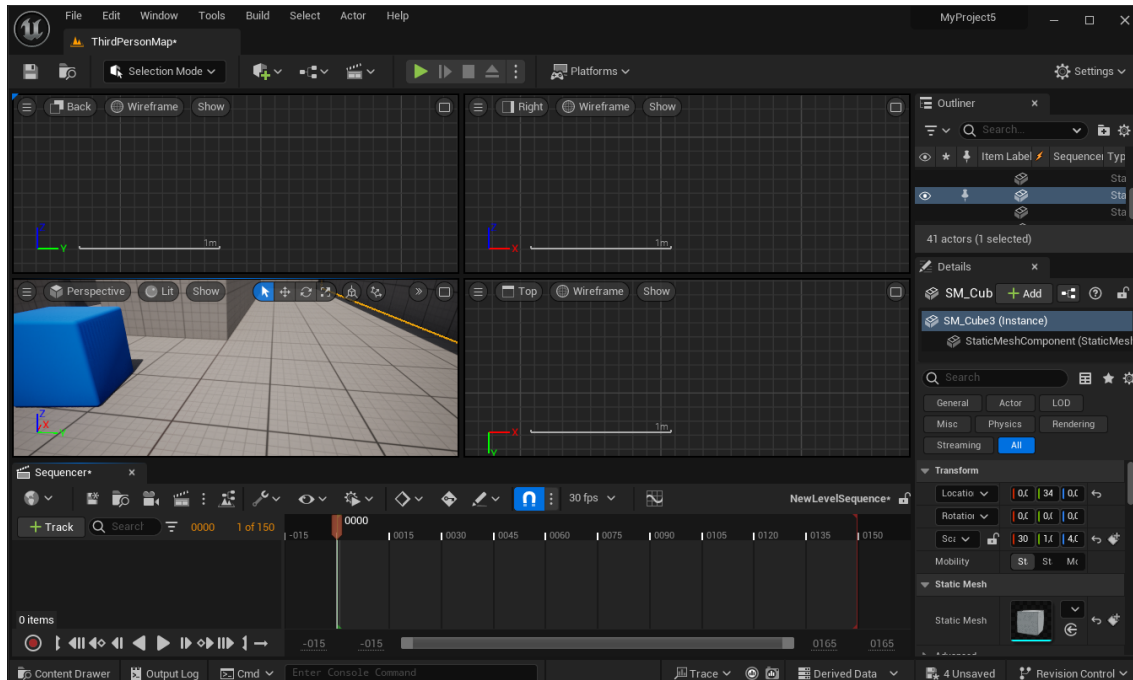
En Secuencer se realizarán secuencias de animación utilizando una interfaz de línea de tiempo similar a la de un software de edición de video. Se puede manipular cámaras, luces, efectos especiales, objetos en movimiento y otros elementos de la escena para crear secuencias cinematográficas complejas y visualmente impresionantes. Podemos separar sus funciones de la siguiente manera.

- Animación de personajes
- Edición no lineal
- Efectos visuales
- Renderizado en tiempo real
- Interpolación de curvas



**Figura 3232**

Sequencer



Nota: En la figura 32 se observa el editor de video conjuntamente con la escena. Elaboración: Autor.

## Autoevaluación 5

1. ¿Qué es un teaser en el contexto de los videojuegos?
2. ¿Cuál es el propósito principal de un teaser en la promoción de un videojuego?
3. ¿Qué elementos suelen incluirse en un teaser para generar interés en los jugadores?
4. ¿Cuál es la duración típica de un teaser de videojuego?
5. ¿Cuál es la diferencia entre un teaser y un tráiler completo de un videojuego?
6. ¿Qué es Sequencer en UE?
7. ¿Puedo crear animaciones en UE5?
8. ¿Tiene editor de video UE5?
9. ¿Cuándo realizo testing?



10. ¿Qué es un empaquetado?

### Resumen de la Unidad 5

La etapa final del proyecto está dedicada al empaquetado y el testing del videojuego. Cuando se obtiene el ejecutable sin errores de compilación se puede comenzar con la etapa de creación del teaser para demostrar todo el proceso de trabajo que se ha realizado a lo largo de la materia.

