



**UNIVERSIDAD POLITÉCNICA SALESIANA**  
**SEDE GUAYAQUIL**  
**CARRERA DE DISEÑO MULTIMEDIA**

**Videojuego 2D ambientado en paisajes y  
mitología Inca en el Ecuador**

**Trabajo de titulación previo a la obtención del  
Título de Licenciada/o en Diseño Multimedia**

**AUTORES: Fernanda Daniela Cobos López y Jean Carlos Pilay Rios**

**TUTOR: Jonatan Andrés Portugal Gorozabel**

**Guayaquil - Ecuador**

**2025**

## **CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN**

Nosotros, Fernanda Daniela Cobos López con documento de identificación N° 0930246772 y Jean Carlos Pilay Rios con documento de identificación N° 0956382840; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, 17 de enero del año 2025.

Atentamente,



---

Fernanda Daniela Cobos López  
0930246772



---

Jean Carlos Pilay Rios  
0956382840

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO  
DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA  
SALESIANA**

Nosotros, Fernanda Daniela Cobos López con documento de identificación No. 0930246772 y Jean Carlos Pilay Rios con documento de identificación No. 0956382840, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto de Titulación: VIDEOJUEGO 2D AMBIENTADO EN PAISAJES Y MITOLOGÍA INCA EN EL ECUADOR, el cual ha sido desarrollado para optar por el título de: Licenciados en Diseño Multimedia, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

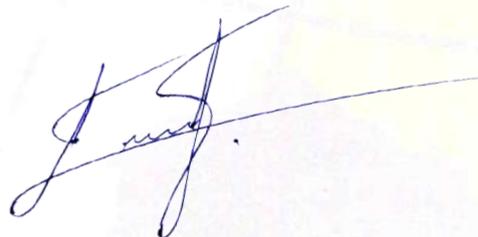
En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 17 de enero del año 2025.

Atentamente,



Fernanda Daniela Cobos López  
0930246772



Jean Carlos Pilay Rios  
0956382840

## CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Jonatan Andrés Portugal Gorozabel con documento de identificación N° 0926631573, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación VIDEOJUEGO 2D AMBIENTADO EN PAISAJES Y MITOLOGÍA INCA EN EL ECUADOR, realizado por Fernanda Daniela Cobos López con documento de identificación N° 0930246772 y por Jean Carlos Pilay Rios con documento de identificación N° 0956382840, obteniendo como resultado final el trabajo de titulación bajo la opción Producto Artístico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 17 de enero del año 2025.

Atentamente,



---

Jonatan Andrés Portugal Gorozabel  
CI. 0926631573

## DEDICATORIA

En primer lugar, le dedico mi proyecto de tesis a mis padres; Eduardo Antonio Cobos López y María Fernanda López Torres. Quienes me enseñaron que con todo esfuerzo y dedicación que le ponga a toda la actividad se puede lograr y mejorar.

A mi hermano, Eduardo José Cobos López, quien estuve para mí en todo momento y me enseñó que muchas veces nuestros propios límites lo ponemos nosotros mismos, gracias de corazón.

A mis abuelos, tío y prima, quienes son parte de mi vida y están siempre presentes sin la necesidad de ser parte de mi núcleo familiar (padres y hermano). A pesar de no verlos todos los días, los pocos momentos que paso junto a ellos siempre llenan de felicidad mi vida.

Finalmente, le dedico también mi proyecto a mis mascotas Snappy y Gris, en especial a Snappy; quien, a pesar de no compartir el mismo idioma, está en cada momento junto a mi lado y me acompañó en cada desvelada que tuve durante mi periodo del colegio, universitario y para realizar el proyecto. Muchas gracias de corazón.

Fernanda Daniela Cobos López

## DEDICATORIA

Le dedico el proyecto a mi familia. Mi padre, Elvis Javier Pilay Merchán y mi madre, Verónica Antonieta Rios Oviedo; quienes son siempre las personas que me apoyan en todo momento y me alientan a seguir adelante a pesar de todos los altos y bajos del día a día, gracias por depositar toda la confianza en mí y demostrarme que con todo esfuerzo y trabajo se puede superar cualquier obstáculo.

Asimismo, le dedico el proyecto a mis hermanos, Dario Javier Pilay Rios y Emely Guadalupe Pilay Rios, quienes son mis compañeros de vida y serán mi mayor apoyo en un futuro, agradezco de todo corazón saber que son ustedes en quienes podré confiar por el resto de mi vida.

Jean Carlos Pilay Rios

## AGRADECIMIENTOS

En primer lugar, quiero expresar mi más profundo agradecimiento a mi familia; quienes me dieron la oportunidad de aprender y estudiar desde temprana edad. Y me motivaron a no dejar o retirarme de la carrera que poco a poco fue teniendo más complejidad.

También agradezco de todo corazón a mi compañero de tesis, Jean Carlos Pilay Rios, quien depositó toda la confianza en mí para poder realizar el trabajo. A pesar de ciertas diferencias y malentendidos durante la elaboración del proyecto, supimos cómo afrontarlo. Gracias Pilay por tenerme paciencia y dedicarle todo tu tiempo y esfuerzo a este proyecto.

Asimismo, agradezco a mis compañeros de promoción que tuvieron la paciencia de conocerme y no juzgarme desde que ingresamos a la universidad. Hoy, puedo decir, que nos vimos crecer como personas y profesionales.

Finalmente, agradezco al Lcdo. Rodrigo Sánchez, quien nos guio en todo momento en la elaboración del apartado de programación. Gracias a todo su conocimiento pudimos finalizar con éxito la creación del videojuego.

Fernanda Daniela Cobos López

## AGRADECIMIENTOS

Quiero expresar mi más sincero agradecimiento a todas las personas que, de una u otra forma, fueron parte fundamental en la realización de este proyecto. A mis amigos y familiares, quienes con su apoyo incondicional y palabras de aliento me motivaron a seguir adelante, incluso en los momentos más desafiantes.

En especial, quiero dedicar este espacio a mi compañera de tesis, Fernanda Daniela Cobos López, cuya confianza en nuestro trabajo fue un pilar fundamental para el desarrollo de este proyecto. Su compromiso, perseverancia y dedicación fueron una fuente de inspiración constante. Este logro no habría sido posible sin su incansable esfuerzo y pasión compartida por alcanzar nuestras metas.

Gracias a todos por ser parte de este camino.

Jean Carlos Pilay Rios

## RESUMEN

El presente proyecto muestra la creación de un videojuego 2D ambientado con los paisajes y mitología Inca en el Ecuador, con un estilo de pixel art. La elaboración de este proyecto busca dar a conocer lugares turísticos del país, y ayudar en la preservación de mitos y cultura Inca en Ecuador, mediante el uso de una herramienta infravalorada que son los videojuegos.

La metodología empleada fue cualitativa, se centra en comprender características culturales y mitológicas; usando el método inductivo, donde se recopila información y detalles concretos sobre personajes del imperio Inca y lugares turísticos del Ecuador.

La investigación se enfoca en explorar y conocer al Dios mitológico del imperio Inca más representativo y lugares donde más se asentaron en Ecuador. Estos elementos son claves para crear personajes atractivos visualmente y lograr una narrativa donde los lugares únicos del Ecuador puedan destacar por su belleza.

El resultado fue un videojuego donde se pueda apreciar con claridad los paisajes del Ecuador en estilo píxel, sin que estos llegasen a perder su esencia. Asimismo, esto contribuirá a la preservación de la cultura y nuestro medio ambiente en el país.

**Palabras claves: (Videojuego, Cultura, Mitología, Ecuador, Pixel art)**

## ABSTRACT

*This project shows the creation of a 2D video game set with Inca landscapes and mythology in Ecuador, with a pixel art style. The development of this project seeks to publicize tourist sites in the country, and help in the preservation of myths and Inca culture in Ecuador, through the use of an undervalued tool that are the video games.*

*The methodology used was qualitative, focused on understanding cultural and mythological characteristics; using the inductive method, where information and concrete details about characters of the Inca empire and tourist sites in Ecuador are collected.*

*The research focuses on exploring and getting to know the most representative mythological gods of the Inca empire and the places where they settled in Ecuador. These elements are key to create visually appealing characters and achieve a narrative where the unique places of Ecuador can stand out for its beauty.*

*The result was a video game where the landscapes of Ecuador can be clearly appreciated in pixel style, without losing their essence. Also, this will contribute to the preservation of culture and our environment in the country.*

**Keywords: Video game, Culture, Mythology, Ecuador, Pixel art**

## ÍNDICE DE CONTENIDO

Contenido	
DEDICATORIA .....	5
DEDICATORIA .....	6
AGRADECIMIENTOS .....	7
AGRADECIMIENTOS .....	8
RESUMEN .....	9
ABSTRACT .....	10
INTRODUCCIÓN .....	14
Problema de estudio .....	14
Justificación .....	16
OBJETIVOS GENERAL Y ESPECÍFICOS .....	18
Objetivos específicos .....	18
FUNDAMENTACIÓN TEÓRICA .....	19
1. Enseñanza por medio de los videojuegos .....	19
1.1. Concepto de edutainment .....	19
1.2. Beneficios de los videojuegos educativos .....	20
1.3. Ejemplos de videojuegos educativos .....	21
1.4 Impacto del aprendizaje basado en videojuegos .....	21
2. Los videojuegos .....	21
2.1. Definición de videojuegos .....	21
2.2. Historia de los videojuegos .....	22
2.3. Clasificación de los videojuegos .....	22

2.4. Impactos social y cultural de los videojuegos .....	23
3. Videojuegos en Ecuador .....	24
3.1. Historia y desarrollo de los videojuegos en Ecuador .....	24
4. Píxel Art .....	25
4.1. Técnicas y estilos .....	25
4.2. Historia y evolución del Píxel Art .....	26
4.3. Aplicaciones modernas del Píxel Art .....	27
4.4. Importancia en la educación y la cultura .....	28
5. Mitología y cultura ecuatoriana .....	29
5.1. Dios mitológico Inca .....	29
6. Cultura del Ecuador .....	29
6.1. Cultura andina .....	30
6.2. Ubicación de las culturas andinas .....	31
6.3. Festivales y celebraciones .....	31
6.4. Música y danza .....	32
METODOLOGÍA .....	33
ANÁLISIS DE RESULTADOS .....	34
PRESENTACIÓN DE HALLAZGOS .....	38
CRONOGRAMA .....	44
PRESUPUESTO .....	45
CONCLUSIONES .....	46
Referencias .....	47
APÉNDICE Y/O ANEXOS .....	51
Anexo 1.- (Recopilación de material visual) .....	51

Anexo 2.- (Selección de lugares, Dios y creación de narrativa).....	54
Anexo 3.- (Diseño y elaboración de animaciones de los personajes, ataques, elementos y escenarios). .....	54
Anexos 4.- (Programación e integración de elementos) .....	61
Anexos 5.- (Elaboración del videojuego - funcionalidad) .....	77
Anexo 6.- (Prototipado - Diseño del Game Document Design del videojuego) .....	79

## INTRODUCCIÓN

### Problema de estudio

Los videojuegos desde su origen siempre han sido considerados únicamente como un medio de entretenimiento y ocio; más no, una manera de compartir conocimientos culturales, especialmente en países no tan desarrollados como Ecuador. Sin embargo, no es adecuado considerar este recurso multimedia tan importante como un solo medio de distracción u ocio, debido a que el desarrollo de estos nos puede brindar múltiples beneficios en diversas áreas tales como la educación, la medicina, entre otros.

Según Guerra y Revuelta (2015), “muchos videojuegos mejoran el nivel de atención, concentración, resolución de problemas y fomentan la creatividad”.

Además, muchas personas tienen en mente que este tipo de instrumentos, los videojuegos, solo deben ser usados por niños y adolescentes, más no por adultos. Es decir, se lo considera como un medio de entretenimiento netamente para jóvenes. Sin embargo, este recurso puede brindar numerosos beneficios en otras áreas, como fomentar el turismo, apoyar el aprendizaje de los estudiantes, entre otros.

En Ecuador, siendo un país donde no se promueve mucho esta actividad como una herramienta de instrucción cultural, ha experimentado numerosas transformaciones en los años recientes. Igual que toda industria, como se mencionó anteriormente, este ha ido cambiando y evolucionando de una manera muy notoria; buscando así poder cambiar la idea o mala reputación que tienen los videojuegos de ser un medio de un solo objetivo, el entretener, y no el de poder ayudar en muchos otros ámbitos.

A lo largo del tiempo se ha notado una disminución en la inculcación sobre la mitología que hay en nuestro país, Ecuador. Sin duda alguna, el compartir y plasmar todo el mundo cultural y riquezas del Ecuador es una forma de mantener viva nuestras raíces por varios años. Ecuador, al ser un país que cuenta con una gran variedad cultura y mitología, es considerado como una nación multiétnica y pluricultural. (Alvarracin, 2015)

La mitología ecuatoriana cuenta una gran variedad de acontecimientos. Sin embargo, esta por lo general solo es contada de generación en generación y este ha ido desvaneciéndose o se ha alterado a lo largo del tiempo, y el ecuatoriano como tal no conoce con exactitud sobre su mitología. (Ponce, 2017)

Hoy en día, se cuenta con una basta variedad de productos multimedia destinados a promover las culturas, las etnias y diversos aspectos de la identidad. Los videojuegos son ejemplos destacados de estos productos. La industria de los videojuegos, al igual que la del cine y otros medios similares, ha tenido un impacto positivo significativo a lo largo de los años.

En Ecuador, hay varios referentes destacados en el ámbito de los videojuegos. Un grupo de ecuatorianos logró desarrollar y compartir un videojuego que llegó a una de las consolas más conocidas a nivel mundial y que actualmente se encuentra disponible en la plataforma Steam. To Leave, nombre del videojuego, fue el primer videojuego creado en Ecuador para la gran industria de Sony, propietaria de la industria de videojuegos PlayStation. El videojuego fue desarrollado por Freaky Creations S.A.; esta empresa fue creada por un grupo de jóvenes entusiastas guayaquileños con el objetivo de hacer crecer la empresa con los años y con varios proyectos a futuro. To Leave, nos narra la historia de Harm, un joven con trastorno maniático depresivo que busca reunir almas del submundo para enviarlas al cielo mediante el uso de tecnología antigua y poder así abrir un portal.

Otro videojuego significativo es Capac Heroes, también creado por desarrolladores ecuatorianos. Este título presenta la historia de civilizaciones como los Incas y los Mayas, que florecieron en el continente americano. Capac Heroes fue creado para dispositivos móviles, estando disponible en la AppStore y en un futuro en la PlayStore. El videojuego explora las aventuras que se revive en diferentes civilizaciones prehispánicas americanas. La historia nos presenta a un rockero carchense, quien se encarga de encontrar personajes históricos y deidades tales como la serpiente emplumada Kukulcán, Atahualpa, entre otros. Además, debe recolectar tesoros mientras se enfrenta con varios enemigos en su aventura; a medida que el jugador va avanzando cada nivel aumenta en dificultad.

## Justificación

Como se mencionó anteriormente, el proyecto propone la creación de un videojuego 2D de estilo arcade, con un diseño pixel art, inspirado y ambientado en los paisajes y mitologías del Ecuador. Con este proyecto se busca generar mayor interés para la preservación cultural de la mitología ecuatoriana y contribuir especialmente en el crecimiento de la industria de videojuegos en el país. Ecuador es un país con una vasta riqueza cultural que incluye una diversidad de mitos, leyendas y paisajes únicos que no han sido explorados en profundidad en los medios interactivos como los videojuegos. Estos elementos constituyen una parte esencial de la identidad nacional, sin embargo, se encuentran en riesgo de ser olvidados. Según Ponce (2017), “la mitología ecuatoriana ofrece una gran variedad de conocimiento, pero la mitología (mitos), que ha sido contada de generación en generación, se ha ido perdiendo a lo largo del tiempo y el ecuatoriano como tal no conoce su mitología”.

Asimismo, la poca industria de los videojuegos que hay en el país no aborda tanto los temas culturales. Aunque títulos como To leave y Samsa y los caballeros de la luz, llegaron a grandes plataformas de videojuegos, estos no exponen temas culturales. Un ejemplo de producto multimedia que aborda temas más generales es Capac Heroes, este videojuego está inspirado en las civilizaciones prehispánicas de América, así como los Incas y los Mayas, sin centralizarse en nuestra cultura como tal.

El gobierno ha mostrado interés apoyando iniciativas de proyectos que busquen afianzar en la cultura ecuatoriana. En el 2023, el Ministerio de Cultura y Patrimonio junto al Instituto de Fomento a la Creatividad, realizaron una convocatoria para participar en concursos, en nueve ámbitos diferentes y cada uno con un monto respectivo; este buscó a varios artistas y gestores culturales del país. Para la producción de videojuegos para dispositivos móviles se invirtió USD\$75.000 a cinco beneficiarios (IFCI, 2023). De esta manera, el estado ya da un incentivo económico que ayudaría motivar a todos los

programadores, diseñadores, estudios independientes, entre otros; en la creación de productos multimedia inspirados en la cultura ecuatoriana.

Finalmente, como señala Jiménez (2022), “los videojuegos son una industria que mueve billones de dólares. Este campo ofrece posibilidades de crecimiento a los desarrolladores ecuatorianos. Sobre todo, porque hay financiamiento disponible”.

En conclusión, la realización de un videojuego 2D ambientado en los paisajes y mitologías del Ecuador es necesaria para potenciar la preservación y difusión de la cultura ecuatoriana, su capacidad para educar de manera innovadora, su oportunidad de mercado y su viabilidad técnica. Este proyecto no solo dará a conocer más el país, sino que también abrirá nuevas posibilidades en la industria de los videojuegos.

## OBJETIVOS GENERAL Y ESPECÍFICOS

Promover el conocimiento y la difusión de la cultura ecuatoriana a través del desarrollo de un videojuego 2D que resalte la riqueza mitológica y paisajística del país.

### Objetivos específicos

1. Investigar las características geográficas, mitológicas, culturales y naturales de varios paisajes del Ecuador que serán representadas en el videojuego.
2. Analizar la mitología y cultura Inca en el Ecuador para identificar los elementos clave que serán incorporados en la narrativa y diseño del videojuego 2D
3. Desarrollar un videojuego 2D ambientado en los diversos paisajes del Ecuador ofreciendo una experiencia visual y lúdica que destaque la riqueza natural y cultural del país.

## FUNDAMENTACIÓN TEÓRICA

### 1. Enseñanza por medio de los videojuegos

Se han realizado investigaciones que destacan los beneficios de los videojuegos como herramientas para la mejora de habilidades sociales y comunicativas. Además, se menciona que estos recursos contribuyen al entrenamiento en trabajo colaborativo, facilitando la asignación de roles y la creación de estrategias para resolver problemas en equipo. De este modo, los estudiantes pueden explorar, investigar, descubrir, dialogar y trabajar en objetivos individuales que llevan a alcanzar un fin común. Estas capacidades y competencias pueden ser promovidas a través de videojuegos que favorecen la participación e interacción entre múltiples jugadores. (León-Atienza et al., 2020)

Los videojuegos favorecen aprendizajes significativos (Roso, 2016), ya que permiten a los jugadores adquirir conocimientos en el área cognitiva mediante la práctica, es decir, "aprender haciendo". Al recrear escenarios de la realidad, los videojuegos facilitan la transmisión de conocimientos a través de la interacción con otros jugadores, objetos y diversas situaciones, posibilitando que lo aprendido pueda aplicarse tanto dentro como fuera del entorno del juego. Además, el uso de videojuegos en contextos específicos estimula el desarrollo del pensamiento lógico (Jiménez-Palacios & Cuenca, 2015), promoviendo la comprensión y el aprendizaje. Este enfoque potencia distintos tipos de conocimiento — conceptual, actitudinal y procedimental— y contribuye a desarrollar habilidades de resolución de problemas. (León-Atienza et al., 2020)

#### 1.1. Concepto de edutainment

##### 1.1.1. Definición de edutainment

El término *edutainment*, que combina las palabras educación y entretenimiento, se refiere a una metodología que busca entretener y educar simultáneamente. Su objetivo es motivar a los estudiantes y facilitar un aprendizaje más efectivo al integrar elementos lúdicos con contenido educativo. Aunque es posible que no estén completamente familiarizados con este término, el concepto de *edutainment* no es nuevo. Incluso aquellos que no lo conocen

lo habrán experimentado en diversas formas; por ejemplo, *edutainment* se puede ver en cuentos como “Érase una vez...”, en programas como “Barrio Sésamo” e incluso en “Dora la exploradora”. Sin embargo, esta metodología ha evolucionado y va más allá de los programas educativos tradicionales, ya que ha encontrado su lugar en el aula gracias a las herramientas y recursos proporcionados por la digitalización educativa, así como la realidad aumentada, la realidad virtual y la inteligencia artificial. Por esta razón, se ha decidido dedicar un artículo de blog a este tema. (Classlife, 2023)

## **1.2. Beneficios de los videojuegos educativos**

Como se mencionó posteriormente, los videojuegos no solo cumplen con un solo fin, sino, estos también pueden brindar momentos de relajación, diversión y potenciar habilidades cognitivas tales como la memoria, atención, motivación, entre otros.

### **1.2.1. Desarrollo de habilidades cognitivas y blandas**

Los videojuegos educativos pueden ayudar a mejorar las capacidades cognitivas, así como la resolución de problemas y la capacidad de entender y analizar una información; de la misma manera las habilidades blandas tales como la cooperación y la comunicación con otros. Los juegos suelen exigir a los jugadores que tomen decisiones estratégicas, trabajen en equipo y adapten sus estrategias para superar obstáculos.

Además, los videojuegos educativos suelen ser más atractivos y motivadores que los métodos tradicionales y obsoletos de enseñanza. La interacción y la gamificación fomentan la participación, lo que puede llevar a una mayor motivación para aprender. Los jugadores están más dispuestos a experimentar y enfrentar desafíos cuando el proceso de aprendizaje se presenta de manera lúdica.

Según Valderrama-Ramo (2012), los videojuegos fomentan la motivación y participación de los usuarios, ya que incorporan elementos que favorecen el desarrollo cognitivo de las personas, aplicando teorías relacionadas con la motivación y el aprendizaje.

### **1.3. Ejemplos de videojuegos educativos**

Minecraft: Education Edition: Utilizado para enseñar conceptos matemáticos, ciencias naturales y habilidades de resolución de problemas. Ofrece un entorno de construcción que fomenta la creatividad y la aplicación práctica de conceptos.

Kerbal Space Program: Enseña principios de física y astronáutica mediante la simulación de un programa espacial, proporcionando una experiencia práctica en la construcción y gestión de naves espaciales.

### **1.4 Impacto del aprendizaje basado en videojuegos**

#### **1.4.1. Desarrollo de habilidades y rendimiento**

Los videojuegos educativos promueven el desarrollo de habilidades de pensamiento crítico mediante la resolución y análisis de problemas complejos y la toma de decisiones estratégicas. Estas habilidades son transferibles a otras áreas del aprendizaje y la vida cotidiana.

El pensamiento crítico puede ser entendido como un tipo de razonamiento complejo y avanzado que exige un enfoque lógico. Implica un conjunto de habilidades, como la formulación de ideas, la capacidad de deducir significados y la evaluación de diferentes argumentos, entre otras. (Almeida & Franco, 2011)

## **2. Los videojuegos**

### **2.1. Definición de videojuegos**

Los videojuegos son una forma de entretenimiento digital interactivo que se juegan a través de dispositivos electrónicos. Estos dispositivos pueden ser consolas de videojuegos, computadoras personales, dispositivos móviles, y más recientemente, plataformas de realidad virtual. Un videojuego típico implica la interacción del jugador con una interfaz gráfica a través de dispositivos de entrada como controles, teclados, ratones, o pantallas táctiles. Los videojuegos combinan gráficos, audio y narrativa para crear experiencias

inmersivas que pueden variar desde simples juegos de rompecabezas hasta complejas historias con mundos abiertos, el cual lo convierte en un producto multimedia.

## **2.2. Historia de los videojuegos**

La historia de los videojuegos comenzó en la década de 1950 y ha evolucionado significativamente hasta la actualidad. Los primeros juegos fueron "Tennis for Two" y "Spacewar" en 1962. La llegada de los videojuegos comerciales se dio en los años 70 con títulos como "Pong", desarrollado por Atari, que popularizó los juegos de arcade. Más adelante, en los 80, la industria experimentó un auge significativo con el lanzamiento de consolas domésticas como la Atari 2600 y juegos icónicos tales como "Pac-Man" & "Donkey Kong". Este período también vio la creación de empresas influyentes como Nintendo y Sega.

La revitalización llegó con la Nintendo Entertainment System (NES) en 1985, que introdujo títulos legendarios como "Super Mario Bros." y "The Legend of Zelda". La década de los 90 fue marcada por la competencia o rivalidad entre Nintendo y Sega, así como la entrada de Sony con la PlayStation, que innovó con el uso de discos compactos en lugar de cartuchos.

El cambio al siglo XXI trajo avances en gráficos, narrativa y jugabilidad. Microsoft se unió a la competencia con la Xbox, y la industria comenzó a explorar juegos en línea y móviles. La última década ha visto un enfoque en la realidad virtual, la realidad aumentada y el creciente fenómeno de los deportes electrónicos (eSports). (Kent, 2001)

## **2.3. Clasificación de los videojuegos**

Los videojuegos se pueden catalogar en varios géneros, incluyendo acción, aventura, rol, simulación, estrategia, deportes, y puzzle. Cada género tiene características distintivas y diferentes formas de interacción y mecánicas de juego. Según (Sedeño Vandellós, 2010) algunas de sus destacadas son:

**Juegos de acción:** Requieren respuestas rápidas y precisas, como disparar o golpear, sin necesidad de planificar.

**Árcade:** Involucran superar niveles con rapidez y atención, ayudando al desarrollo psicomotor y la orientación espacial.

**Juegos de estrategia:** Enfatizan la planificación y la resolución de problemas, desarrollando el pensamiento lógico y la organización mental.

**Juegos de aventura:** Se centran en la interactividad y la toma constante de decisiones.

**Juegos deportivos:** Involucran jugadores reales y la gestión de equipos, y requieren habilidades, rapidez y precisión.

**Juegos de simulación:** Permiten experimentar y controlar situaciones complejas, desarrollando conocimientos específicos y estrategias.

**Juegos de rol:** Basados en la evolución de personajes, fomentan el cálculo mental, el vocabulario y habilidades sociales, así como la empatía y el trabajo en equipo.

**Juegos masivos (MMORPG):** Permiten a miles de jugadores interactuar en un mundo virtual simultáneamente.

**Juegos de supervivencia:** El objetivo es sobrevivir enfrentando situaciones o enemigos, a menudo mezclándose con otros géneros como el terror.

Estos juegos no solo entretienen, sino que también pueden ser herramientas educativas y de desarrollo de habilidades.

## 2.4. Impactos social y cultural de los videojuegos

Los videojuegos han influido en la cultura popular y se han convertido en una forma de arte. Las narrativas, el diseño gráfico y la música en los videojuegos reflejan y afectan las

tendencias culturales. Además, se puede destacar que este ya tiene su propiedad comunidad de usuarios “gamers”.

Los videojuegos influyen en la socialización, la educación, y el comportamiento. Pueden fomentar habilidades como la solución de problemas y el trabajo en equipo, pero también han sido criticados por potencialmente promover la violencia o el aislamiento.

Los videojuegos han evolucionado como reflejo de problemas políticos y sociales, destacando Animal Crossing: New Horizons como una plataforma de reivindicación social y política, especialmente durante la COVID-19 y las protestas en Hong Kong. (Moreno Cantano, 2020)

### **3. Videojuegos en Ecuador**

#### **3.1. Historia y desarrollo de los videojuegos en Ecuador**

##### **3.1.1. Orígenes tempranos**

Los videojuegos en Ecuador han comenzado a desarrollarse de manera significativa en la última década. Los primeros videojuegos ecuatorianos aparecieron en la década de 2000, y la comunidad de desarrolladores ha crecido con la disponibilidad de herramientas de desarrollo más accesibles y la promoción de eventos locales.

El sector de desarrollo de videojuegos se ha consolidado como una parte clave del mercado global del entretenimiento. Esto respalda la inversión en la creación de nuevos proyectos dentro del campo y la generación de empleos para quienes trabajarán como desarrolladores de videojuegos. (Jiménez Cajamarca & Vásquez Badillo, 2019)

A medida que la tecnología se volvió más accesible, comenzaron a surgir estudios de desarrollo y videojuegos independientes. La creación de eventos y competiciones locales, como Game Jams, proporcionó una plataforma para que los desarrolladores novatos mostraran su trabajo. Estos eventos ayudaron a construir una comunidad de desarrolladores y fomentaron la colaboración entre ellos.

### **3.1.2. Crecimiento reciente**

En la última década, la industria ha experimentado un crecimiento notable. El acceso a herramientas de desarrollo más avanzadas y a internet ha permitido que los desarrolladores ecuatorianos creen videojuegos de calidad superior. La participación en ferias internacionales y el reconocimiento en competencias globales son testimonio de este crecimiento.

En Ecuador, un grupo de jóvenes compatriotas lograron desarrollar un videojuego y darlo a conocer a todos los usuarios en una de las plataformas de distribución digital de videojuegos más famosas del mundo, Steam. Freaky Creations es un estudio de desarrollo de videojuegos independiente. En el 2018, logró una gran meta que muchos ecuatorianos lo ven inalcanzable; “To Leave” fue lanzado al mercado, inicialmente en la PlayStation Store. Hoy en día, se encuentra disponible en Steam. Este videojuego trata sobre como Harm, el personaje, quien es un maniático depresivo que busca reunir almas del submundo para enviarlas al cielo mediante una puerta que él posee.

Asimismo, se pueden mencionar otros videojuegos ecuatorianos lanzados al mercado tales como “Capac: Heroes”, “Samsa y los Caballeros de La Luz” y “El Gran viaje”, que no se la ha dado la cobertura necesaria sobre su producción y desarrollo. (Jiménez Cajamarca & Vásquez Badillo, 2019)

En Ecuador, existen desarrolladores altamente capacitados y una gran cantidad de personas apasionadas por los videojuegos. Se considera que el futuro de la industria local tiene un gran potencial. (IFCI, 2018)

## **4. Pixel Art**

### **4.1. Técnicas y estilos**

El pixel art se caracteriza por el uso de técnicas específicas, como el dithering, que consiste en mezclar dos colores para crear la ilusión de un tercer color, y el antialiasing, que suaviza los bordes de las formas para evitar una apariencia escalonada. Estas técnicas

permiten a los artistas crear imágenes detalladas y complejas, a pesar de las limitaciones inherentes al medio. Hay dos tipos principales de pixel art: el isométrico y el no isométrico.

**Pixel Art Isométrico:** Esta técnica utiliza ángulos específicos para crear una apariencia tridimensional en un entorno bidimensional. En otras palabras, el arte parece tener profundidad y volumen, aunque se visualiza en un plano 2D. Ejemplos de juegos que emplean esta técnica son *Golden Sun*, *Final Fantasy Tactics* y *Super Mario RPG*. (shirani, 2017)

**Pixel Art No Isométrico:** Esta técnica consiste en proyectar la imagen de forma plana, ya sea desde una vista superior, lateral o desde abajo. Es el estilo que se usa en la mayoría del pixel art, como en *Super Mario Bros*, *Pokémon* y *Castlevania*.

Aunque son técnicas diferentes, a veces se combinan en un mismo juego o dibujo, mostrando, por ejemplo, personajes no isométricos en un entorno o zonas jugables que sí utilizan la técnica isométrica. (shirani, 2017)

Además, los artistas de pixel art suelen trabajar con una paleta de colores limitada, lo que requiere una cuidadosa consideración del uso del color. La elección de colores y la forma en que se combinan pueden tener un gran impacto en la apariencia y la sensación de la obra final.

## 4.2. Historia y evolución del Pixel Art

El pixel art no fue creado por una sola persona, sino que se desarrolló con el tiempo a medida que la tecnología de los ordenadores avanzaba. La primera imagen digital, que se considera un precursor del pixel art, fue creada en 1957 por el científico informático estadounidense Russel Kirsch. Esta imagen, que muestra a su hijo recién nacido, está compuesta por 512 píxeles. Durante los años 70, la llegada de ordenadores con pantallas de mayor resolución permitió crear imágenes más detalladas y complejas. En esa década, aparecieron los primeros programas de edición de imágenes, como SuperPaint, que facilitaron la creación de pixel art. El pixel art alcanzó gran popularidad en los años 80

gracias a los videojuegos, que usaron esta técnica para sus gráficos. Juegos icónicos de la época, como Space Invaders, Pac-Man y Super Mario Bros, están hechos en pixel art (U-tad, 2023).

Con la evolución de la tecnología y el aumento en la capacidad gráfica de las consolas y computadoras, el pixel art fue gradualmente reemplazado por gráficos más avanzados. Sin embargo, el estilo nunca desapareció por completo y ha experimentado un renacimiento en la última década, especialmente en la escena de los videojuegos indie.

### 4.3. Aplicaciones modernas del Pixel Art

Existen diversos programas gratuitos para crear pixel art, y la elección del adecuado dependerá de varios factores, principalmente la facilidad de uso, las herramientas disponibles y el costo del software. (Carranza, 2022)

- **Aseprite:** Este software destaca por su interfaz sencilla, de fácil entendimiento y herramientas intuitivas. Permite crear animaciones y dibujos de píxeles simétricos, facilitando la creación de diseños detallados.
- **Piskel:** Ideal para crear imágenes animadas de píxeles, Piskel permite añadir nuevas capas (frames) para diseñar secuencias de animación para tus personajes.
- **GIMP:** Con funcionalidades similares a las de Photoshop, GIMP es una herramienta gratuita que es excelente para retocar y crear fotomontajes de tus dibujos en pixel art.
- **Krita:** Disponible para Windows, Mac y Linux, Krita es un programa gratuito y fácil de usar para diseñar pixel art. También es útil para otras formas de ilustración.
- **Pixilart:** Esta plataforma online permite crear pixel art y animaciones, con opciones para ajustar el brillo de los píxeles y ver el trabajo de otros usuarios para inspirarte.

- **Clip Studio Paint:** Facilita la creación de pixel art mediante plantillas de 1x1 y pinceles especiales. También es una excelente herramienta para crear cómics.
- **Illustrator:** Aunque es conocido por el diseño gráfico, Illustrator también se puede usar para crear pixel art, especialmente si ya estás familiarizado con el programa.
- **Adobe Photoshop:** Photoshop, uno de los programas de retoque de imágenes 2D más populares, te permite crear animaciones y dibujos en pixel art de manera eficiente.
- **Pixel Studio:** Esta aplicación para dispositivos móviles y tabletas gráficas permite crear pixel art y ha sido descargada por millones de usuarios en todo el planeta. Incluye funciones como autoguardado y lapso de tiempo para animaciones.
- **Dotpict:** Similar a Pixilart, Dotpict ofrece una interfaz con texto e íconos en píxeles. Destaca por su función de autoguardado y la capacidad de guardar animaciones GIF de tu proceso de dibujo.
- **Pixaki:** Una aplicación con una interfaz multitáctil y paletas personalizables. Soporta hasta 50 capas y animaciones, y es ideal para desarrolladores de videojuegos que necesitan exportar hojas de sprites.
- **Procreate:** Aunque es una aplicación de pintura digital para iPad, Procreate también permite crear pixel art. Su función Palette Capture facilita la creación de paletas de colores personalizadas.
- **Pyxel Edit:** Popular entre los artistas de píxeles por su interfaz fácil manejo, Pyxel Edit ofrece una versión gratuita, aunque la versión completa incluye más funciones útiles.

#### 4.4. Importancia en la educación y la cultura

El pixel art, que usa pequeños bloques de color llamados píxeles, ha trascendido de los videojuegos a la educación. Esta técnica fomenta la creatividad y el pensamiento crítico al permitir a los estudiantes experimentar con formas y patrones en un espacio limitado.

Además de desarrollar habilidades artísticas, el pixel art ayuda a entender conceptos matemáticos y científicos, como geometría y proporciones. Su naturaleza digital facilita la creación y compartición a través de herramientas en línea, eliminando la necesidad de materiales costosos. (Lauren, 2024)

### **¿Cómo se usa el pixel art en el aula?**

El pixel art puede enseñar creatividad y geometría, además de ofrecer una perspectiva histórica sobre los videojuegos. También promueve habilidades de resolución de problemas al planificar y manipular píxeles. Es una herramienta educativa versátil que combina arte, matemáticas y lógica. (Lauren, 2024)

## **5. Mitología y cultura ecuatoriana**

La mitología ecuatoriana es rica y diversa, reflejando la vasta variedad de culturas indígenas que habitan el país. Los mitos y leyendas son una parte integral de la herencia cultural de Ecuador y proporcionan una visión única de las creencias y valores de sus pueblos.

### **5.1. Dios mitológico Inca**

#### **Inti - Dios del Sol**

Inti es el Dios del Sol en la mitología Inca y es considerado uno de los dioses más importantes. Inti es visto como el dador de vida y el protector de los Incas. Los emperadores incas, considerados descendientes de Inti, llevaban a cabo ceremonias y festivales en su honor, como el Inti Raymi, una celebración del solsticio de invierno. (Mythical Encyclopedia, s.f.)

## **6. Cultura del Ecuador**

Ecuador es un país diverso y multicultural, con una rica y vasta herencia donde se evidencia y refleja la influencia de las culturas indígenas, africanas y europeas. Esta

diversidad cultural se manifiesta en sus tradiciones, costumbres, festivales, música, danza, artesanía y gastronomía.

### **6.1. Cultura andina**

Las civilizaciones andinas fueron culturas precolombinas que florecieron en el área occidental de Sudamérica, principalmente las más cercanas de la cordillera de los Andes, a lo largo de casi dos mil años, hasta la llegada de los españoles en el siglo XVI.

Este territorio constituyó una región cultural rica y muy diversa, considerada una de las cunas de la civilización humana, ya que allí se desarrollaron sociedades complejas de forma independiente al resto del mundo.

Se estima que las primeras culturas andinas aparecieron entre el 9,000 y el 5,000 a.C., en el período paleolítico. La comprensión profunda de la tradición andina resulta difícil debido a la ausencia de textos escritos previos a la llegada de los conquistadores. Cada hallazgo arqueológico no solo agrega información, sino que con frecuencia obliga a replantearse lo que se conoce sobre estas antiguas sociedades. Existen tres teorías principales sobre el origen de estas culturas, cada una apoyada por especialistas que presentan diferentes perspectivas sobre el funcionamiento de estas civilizaciones prehispánicas.

**La teoría difusionista, planteada por el arqueólogo Max Uhle (1856-1944),** sugiere que la civilización andina se originó en la costa y luego se extendió hacia las zonas montañosas. De acuerdo con esta perspectiva, los pueblos pesqueros primitivos de la región peruana habrían sido impulsados a organizarse gracias a una influencia ancestral procedente de Mesoamérica.

**La teoría autoctonista de Julio C. Trello (1880-1947),** propuso que el origen de esta cultura se encuentra en la selva amazónica peruana, donde la cultura Chavín habría sido la primera en desarrollarse para luego expandirse hacia las áreas vecinas. Esta perspectiva rechaza cualquier influencia externa; sin embargo, algunos otros defensores de la teoría autóctona consideran probable que el origen de estas civilizaciones sea más bien costero.

**La teoría aloctonista de Federico Kauffman Doig (1928)**, sugiere que el origen de la cultura andina se encuentra en la costa de Ecuador, específicamente en la cultura Valdivia, considerada el punto de partida más probable para el desarrollo de estas civilizaciones.

En cualquier caso, distintas perspectivas sobre las culturas andinas concuerdan en que su época de mayor esplendor y crecimiento se dio entre los siglos I y IX d.C.

## 6.2. Ubicación de las culturas andinas

Las culturas andinas se asentaron y extendieron principalmente a lo largo de la cordillera de los Andes; lo que hoy en día es el actual territorio de Argentina, Chile, Bolivia, Perú, Ecuador, Colombia y en menor rango en la región oeste de Venezuela.

## 6.3. Festivales y celebraciones

Ecuador celebra una variedad de festivales a lo largo del año, muchos de los cuales están relacionados con eventos religiosos, agrícolas y culturales.

### **Inti Raymi**

El Inti Raymi, o también conocido como la Fiesta del Sol, es una celebración Inca que se festeja durante el solsticio de invierno. Es particularmente importante en la región andina y se celebra con danzas, música y rituales en honor al Dios del Sol, Inti. Esta festividad es un reflejo de la continuidad de las tradiciones indígenas en la vida moderna. (El hotele cuatresures quito, 2024)

### **Carnaval**

El Carnaval es una de las celebraciones más grandes de Ecuador, marcada por desfiles, fiestas y la tradicional "fiesta del agua", donde la gente se arroja agua entre sí. En la ciudad de Ambato, el Carnaval se celebra con la Fiesta de las Flores y las Frutas, que incluye desfiles de carrozas decoradas con flores y frutas, así como danzas y conciertos. (El hotele cuatresures quito, 2024)

## **Día de los Difuntos**

El Día de los Difuntos es una festividad en la que las familias ecuatorianas honran a sus seres queridos fallecidos. Las celebraciones incluyen la preparación de alimentos tradicionales como la colada morada y las guaguas de pan, que son ofrecidos en los cementerios y en los hogares. (El hotele cuatresures quito, 2024)

## **6.4. Música y danza**

La música y la danza son componentes vitales de la cultura ecuatoriana, con una variedad de estilos que reflejan la diversidad del país.

### **Marimba**

La marimba es un instrumento musical y un género de música que es particularmente significativo en la región costera de Esmeraldas, donde es interpretada principalmente por la comunidad afroecuatoriana. La marimba y sus ritmos son una parte fundamental de las celebraciones y festivales locales. (patrimoniocultural, 2003)

### **Pasillo**

El pasillo es un género musical que se considera la "música nacional" de Ecuador. Es un tipo de balada que generalmente se interpreta con guitarra y a menudo contiene letras melancólicas. Es popular en todo el país y ha producido algunos de los músicos y compositores más famosos de Ecuador. (patrimoniocultura, s.f.)

## METODOLOGÍA

### **Tipo de investigación: cualitativa**

La presente investigación es de tipo cualitativa porque se centró en explorar y comprender características culturales y mitológicas del imperio Inca en el Ecuador. No se enfoca en números, sino en retratar detalles que sean atractivos visualmente en base a los paisajes del país.

### **Método: Inductivo**

En el proyecto se empleó el método inductivo, ir de lo específico a lo general. Se empezó a recopilar información y detalles concretos sobre personajes mitológicos para realizar una guía del desarrollo del videojuego (apariencia, habilidades y lugar de ubicación). Además, se generaron más ideas para poder integrar la narrativa y el diseño del videojuego para que concuerde con las características y habilidades de los dioses mitológicos.

### **Técnica: Revisión bibliográfica**

Investigación Documental: Revisión de textos, artículos y recursos digitales sobre mitologías y paisajes ecuatorianos.

Desarrollo Inductivo: A partir de la revisión, se generan principios generales para la creación de elementos visuales y narrativos del videojuego.

### **Aplicación de los resultados:**

Los hallazgos se aplican al diseño del juego, garantizando una representación auténtica y educativa de la cultura ecuatoriana. Se hace uso del estilo pixel art para representar los paisajes ecuatorianos sin llegar a perder su esencia o encanto natural. Además, se incorpora una mini historia para que el jugador pueda leer información sobre los paisajes y los dioses mitológicos.

## ANÁLISIS DE RESULTADOS

La elaboración de un videojuego 2D ambientado en paisajes y mitología Inca en el Ecuador representó un reto, dado que su objetivo fue promover el conocimiento y la difusión de la cultura ecuatoriana, resaltando su riqueza mitológica y paisajística del país. Este objetivo se logró a través de un diseño que integra todos los elementos visuales, paisajes y personajes en pixel art, junto con una narrativa que agrega valor al producto.

Durante el desarrollo del proyecto, se realizaron varias revisiones bibliográficas a distintos artículos científicos, revistas, tesis elaboradas por egresados en carreras afines, entre otros. La revisión bibliográfica permitió detallar las siguientes respuestas:

OBJETIVOS ESPECÍFICOS	HERRAMIENTAS	RESULTADOS
Investigar las características geográficas, mitológicas, culturales y naturales de varios paisajes del Ecuador que serán representados en el videojuego.	<ul style="list-style-type: none"> <li>• Visita a la biblioteca Municipal de Guayaquil.</li> <li>• Revisión de revistas.</li> <li>• Investigación y análisis de imágenes recopiladas.</li> </ul>	<p>R1: Ecuador cuenta con una viva riqueza de paisajes únicos.</p> <p>R2: Selección de tres lugares turísticos para ser representados en el videojuego.</p>
Analizar la mitología y cultura Inca en el Ecuador para identificar los elementos clave que serán incorporados en la narrativa y diseño del videojuego 2D.	<ul style="list-style-type: none"> <li>• Revisión bibliográfica</li> <li>• Investigación del Imperio Inca en el Ecuador.</li> <li>• Investigación de dioses Incaicos.</li> </ul>	<p>R1: Conocer los lugares donde se asentaron los Incas en el Ecuador.</p> <p>R2: Selección del Dios Inca más conocido de la cultura.</p>

<p>Desarrollar un videojuego 2D ambientado en los diversos paisajes del Ecuador ofreciendo una experiencia visual y lúdica que destaque la riqueza natural y cultural del país.</p>	<ul style="list-style-type: none"> <li>• Lluvia de ideas.</li> <li>• Aseprite.</li> <li>• Bocetos de paisajes y personajes.</li> <li>• Unity</li> </ul>	<p>R1: Representación más cercana a los paisajes del Ecuador y diseño de personajes en estilo pixel art.</p>
---	---	--

Como se mencionó anteriormente, parte de la obtención de los resultados se realizó mediante varias revisiones bibliográficas y visitas a la biblioteca municipal para obtener más información de lugares turísticos del Ecuador. En consecuencia, se seleccionaron tres lugares turísticos del país que destacan por su belleza natural. Estos son: la Mitad del Mundo, ubicado en Quito; la Laguna de Quilotoa, ubicado en Pujilí, Cotopaxi; y por último las Ruinas de Ingapirca, ubicado en la provincia de Cañar.

Siguiendo con la investigación en mayor profundidad, dos de los lugares seleccionados, Quito y Cañar, o como también se los conocía en ese entonces Kitu (Quito) y Tumi Pamba (actualmente Cuenca); fueron los principales sitios donde se asentó el pueblo incaico.

Las Ruinas de Ingapirca fue un centro ceremonial y observatorio solar, por ende, también se lo conocía como el Templo del Sol. Por otra parte, Quito fue el otro sitio principal de los Incas por su ubicación geográfica, la cual ayudaba en su objetivo de consolidar el Ecuador. También, se debe mencionar que ambos lugares les beneficiaban al Imperio Inca por tener abundancia de recursos naturales. Por otro lado, la Laguna de Quilotoa es un lugar turístico visitado por muchos extranjeros, dado que, los colores turquesas y azules del lago destacan mucho y es muy atractivo visualmente.

Asimismo, conocer más sobre la cultura Inca ayudó saber sobre sus dioses tales como Viracocha, Cocha Mama, Pacha Mama, entre otros. Sin embargo, existe el nombre de un

Dios que es el más conocido entre las personas, hablamos del Dios Inti. Él era el Dios del Sol, el todopoderoso y benevolente; los incas también lo simbolizaban como vida, prosperidad y tiempo.

Luego, se seleccionaron las ubicaciones para los niveles del videojuego y un Dios representativo como enemigo, sin mencionar otros personajes que también se necesitaba integrar como parte de la narrativa del videojuego. Para la elaboración de estos, fue necesario simplificar cada uno de sus elementos sin que este pierda su esencia.

Por parte de los gráficos de cada nivel, se utilizaron colores relacionados con los sitios seleccionados, o algunos que ayudaran a destacar su elemento principal. Para la Mitad del Mundo, se escogieron colores como marrón rojizo en el monumento, puesto que, con esos colores ayudaba a destacar más con todo alrededor. Se usaron verdes para la representación de la vegetación y tonos azules para el cielo despejado. Quilotoa, se representó con tonos de azul brillante para que, de la sensación de agua cristalina, junto a marrones cálidos de las montañas volcánicas que le rodeaban.

Adicionalmente, se realizó el diseño del Dios Inti. Inti era el Dios del Sol. Por ende, se usaron colores amarillos en su vestimenta para representar su relación con el sol y se simplificó su apariencia sin que perdiera su esencia principal: la esencia de un Dios todopoderoso.

Finalmente, después de realizar la investigación necesaria para adaptar los elementos, paisajes y personajes seleccionados para la narrativa del videojuego, se procedió a la creación de todo lo mencionado mediante un programa de paga llamado Aseprite.

Aseprite, es un programa que ayuda en la elaboración de gráficas o ilustraciones en estilo pixel art, tanto para realizar un recurso visual inactivo o que este cuente con algún tipo de animación; como fue en el caso para la creación de los movimientos y ataques de los personajes.

En el apartado de desarrollo, la elaboración del videojuego se logró gracias al motor de desarrollo Unity. Este es un software de desarrollo de videojuegos, ya sea 2D o 3D; considerado entre uno de los software más versátiles y populares para la creación de videojuegos en el mundo. Con la ayuda de este software, se completó un proyecto funcional: un juego plataformero con tres niveles, donde el usuario pueda hacer acciones básicas como correr, saltar y esquivar; y también logra visualizar paisajes del Ecuador.

Con todos los resultados obtenidos durante la elaboración del proyecto, se puede destacar que el pixel art representa bien los paisajes ecuatorianos, haciendo uso de colores y siendo fiel a las estructuras y composición del paisaje. Aunque el estilo sea sencillo, los jugadores podrán identificar elementos culturales y naturales del Ecuador solo con los detalles visuales básicos. También, mediante una apropiada investigación se puede integrar personajes, como el Dios Inti, quien fue uno de los dioses más representativos de la cultura Inca.

De igual manera, la elaboración de un juego inspirado en la cultura y mitología Inca puede ser clave para que los jugadores descubran la información de forma interactiva y progresiva. Un videojuego bien diseñado puede ser una herramienta educativa, ya que permite que los jugadores aprendan de forma visual, conociendo más sobre los paisajes y mitologías del Ecuador. En un contexto escolar, podría motivar a los estudiantes a conocer más sobre nuestra cultura mientras juegan, convirtiendo el aprendizaje en algo entretenido y significativo.

## PRESENTACIÓN DE HALLAZGOS

Durante todo el proceso en la elaboración del videojuego, se destacaron varios hallazgos. Para la creación del proyecto primero se indagó e investigó en revistas académicas, revisiones bibliográficas de artículos científicos, artículos digitales, recursos digitales, entre otros, los cuales ayudaron a conocer más sobre la naturaleza que posee Ecuador.

La mayor fuente de ayuda en la recopilación de recursos audiovisuales fue internet. Conocer cómo lucen los sitios es fundamental para mantener su estructura y este sea reconocible a la hora de diseñar un bocetado del mismo. Ver anexo 1 (recursos visuales).

Además, se analizaron elementos claves de la cultura Inca en el Ecuador; identificar los sitios donde más se asentó el Imperio Inca en el país es fundamental al momento de selección de los niveles desarrollados. Conocer un poco más estos lugares a detalle nos ayuda a representar de manera clara y fiel los paisajes seleccionados para cada nivel del videojuego.

En esta ocasión, se seleccionó dos lugares: Kitu (Quito) y Tome Pamba (actualmente Cuenca). También, se buscó sobre el Imperio Inca. Averiguar cuáles eran los dioses que ellos adoraban, ayudaría al momento de seleccionar un Dios para la narrativa del videojuego; saber cómo incorporarlo y diseñarlo. Durante la investigación se observó la gran relevancia que tiene del Dios Inti para el imperio Inca, o también conocido como Dios del Sol. Se debe destacar que uno de los recursos que más ayudó fueron imágenes y vídeos, ya que conocer cómo era el Dios Inca y ver como son los paisajes ayuda a tener una mejor idea para el bocetado de los elementos mencionados. Ver anexo 1 (recursos visuales).

Una vez definidos los elementos narrativos del videojuego, fue necesario seleccionar la mecánica principal del juego. Se seleccionó la mecánica de exploración. Esta mecánica ayudó a integrar de manera efectiva la narrativa junto a la información de los sitios de cada nivel. La interacción del jugador dentro del nivel y los desafíos del juego brinda una experiencia educativa y de entretenimiento. Para poder integrar todos los elementos antes

mencionados se realizó una breve historia que da la introducción al usuario al momento de comenzar el videojuego. En esta se expresa lo siguiente:

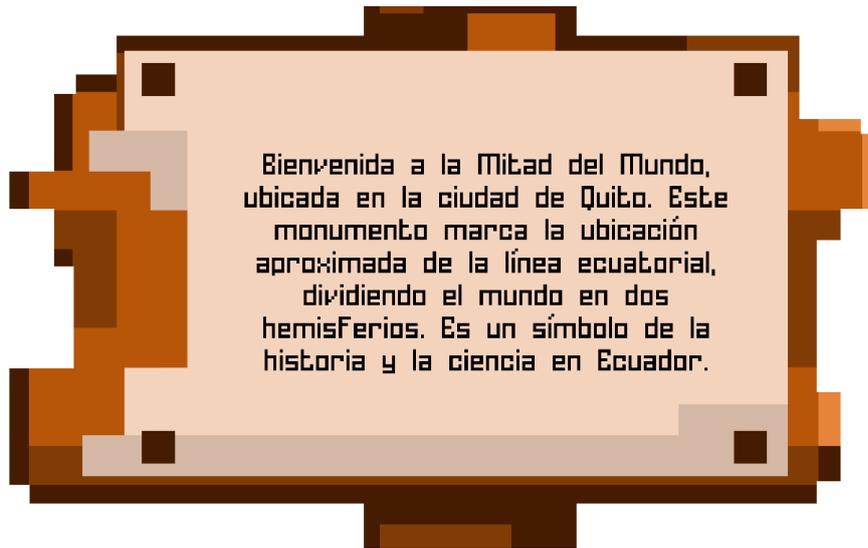
*-Se muestra en pantalla una introducción a la aventura de la protagonista, Amaru-*

*En un pueblo escondido entre los majestuosos Andes ecuatorianos vivía Amaru, una joven curiosa y valiente. Una noche, una luz misteriosa apareció en su ventana, guiándola hacia un rincón remoto del bosque. Allí, en medio de la nada, encontró una espada brillante clavada en una piedra, irradiando una energía que resonaba con su interior. Sin entender por qué y llena de curiosidad decidió tomarla, desatando sin querer una antigua maldición.*

*Esa espada, conocida como Kawsaykuna, liberó a un Dios Inca corrompido por el mal: Inti, el Dios del Sol; y este comenzó a sembrar el caos.*

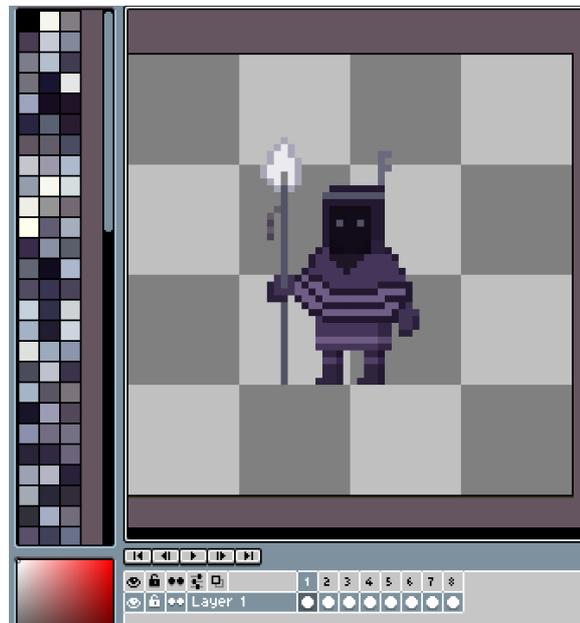
*Con la espada en mano, Amaru no tuvo otra opción que enfrentar lo desconocido. Guiada por un misterioso espíritu de luz, inició un viaje que la llevaría a lugares emblemáticos de su tierra: la Mitad del mundo, la majestuosa laguna de Quilotoa y las ruinas de Ingapirca. Cada paso de su aventura será una prueba de su valor.*

Además, se recopiló información de los sitios a usar en cada uno de los niveles del videojuego. La información se redactó de manera corta pero concisa, con el objetivo de que este le brinde cierto conocimiento al jugador sin que llegue a ser tedioso en la lectura. Ver anexo 3 (diseño de elementos).



*Figura 1.* Cartel informativo sobre la Mitad del Mundo, Quito.

Finalmente, teniendo toda la información y recursos necesarios para la elaboración de la narrativa y diseño de los paisajes y personajes, se procedió generar una idea más limpia y clara de cada personaje y escenario, es decir, se necesitó de elementos finales y que estos dejen de ser bocetos. Ver anexo 3 (diseño de personajes).



*Figura 2.* Diseño del enemigo genérico, Idle.



*Figura 3.* Diseño del enemigo genérico, Ataque.

Para la elaboración del enemigo genérico se tomó como referencia visual a una persona perteneciente al ejército del imperio Inca. Ver anexo 1 (recursos visuales).

Además, para darle más vida y dinamismo al videojuego se incorporó ataques a los personajes y animaciones para que estos no sean solamente observados como elementos inanimados dentro del entorno. Ver anexo 3 (diseño de elementos).



*Figura 4.* Diseño de elemento, ataque del jefe Inti.

Del mismo modo, una vez obtenidos todos los diseños necesarios para integrarlos a la narrativa del videojuego, se procedió a desarrollar la jugabilidad. Por ende, se hizo uso del software Unity, este es conocido mundialmente por brindar versatilidad en la creación de un videojuego 2D o 3D. Para poder darle vida a cada uno de los personajes se elaboró

diversos scripts que facilitó en la organización de estas mismas; separando por acciones cada uno de los scripts, para darle dinamismo al personaje principal, Amaru, se creó scripts de acciones básicas como correr, agacharse, saltar, caer, entre otros. Ver anexo 4 (programación).

```

    animator = GetComponent<Animator>();
    rb = GetComponent<Rigidbody2D>();

    if (boxCollider == null)
    {
        boxCollider = GetComponent<BoxCollider2D>();
    }

    posicionInicio = transform.position;
}

void Update()
{
    // Si las entradas están bloqueadas, no hacer nada
    if (entradasBloqueadas || Time.timeScale == 0 || movimientoBloqueado || enAnimacionDeDaño)
    {
        return;
    }

    enSuelo = Physics2D.Raycast(transform.position, Vector2.down, distanciaSuelo, capaSuelo);

    if (rb.linearVelocity.y > 0 && !enSuelo)
    {
        estaSaltando = true;
        animator.SetBool("isJumping", true);
        animator.SetBool("isFalling", false);
    }
    else if (rb.linearVelocity.y < 0 && !enSuelo)
    {
        estaSaltando = true;
        animator.SetBool("isJumping", false);
        animator.SetBool("isFalling", true);
    }
    else if (enSuelo)
    {
        estaSaltando = false;
        animator.SetBool("isJumping", false);
        animator.SetBool("isFalling", false);
    }
}

```

Figura 5. Códigos de programación: salto de Amaru.

```

Assets > scripts > ControlPersonajes
1 using UnityEngine;
2 using TMPPro;
3
4 public class ControlPersonaje : MonoBehaviour
5 {
6     public float velocidad = 5f; // Velocidad de movimiento
7     public float fuerzaSalto = 10f; // Fuerza del salto
8     public float distanciaSuelo = 0.1f; // Distancia para detectar el suelo
9     public LayerMask capaSuelo; // Capa del suelo para el Raycast
10    public Vector3 posicionInicio; // Posición inicial del personaje
11    public ControlVidas controlVidas; // Referencia al ControlVidas para manejar las oportunidades
12
13    public BoxCollider2D boxCollider; // Referencia al BoxCollider2D del personaje
14    public Vector2 tamañoNormal = new Vector2(1f, 2f); // Tamaño del collider cuando está de pie
15    public Vector2 tamañoAgachado = new Vector2(1f, 1f); // Tamaño del collider cuando está agachado
16    public Vector2 offsetNormal = new Vector2(0f, 0f); // Offset del collider cuando está de pie
17    public Vector2 offsetAgachado = new Vector2(0f, -0.5f); // Offset del collider cuando está agachado
18
19    private Animator animator; // Referencia al Animator
20    private Rigidbody2D rb; // Referencia al Rigidbody2D
21    private bool enSuelo = true; // Verifica si está en el suelo
22    private bool estaSaltando = false; // Verifica si está saltando
23    private bool estaAgachado = false; // Verifica si está agachado
24    private bool enAnimacionDeDaño = false; // Indica si está en la animación de daño
25    private bool movimientoBloqueado = false; // Indica si el movimiento está bloqueado
26    private bool entradasBloqueadas = false; // Bloquea todas las entradas, incluidos clics y teclas
27    public int vidaActual = 3; // Vida actual del personaje
28
29    public CambioEscenaPorEliminacion cambioEscena; // Referencia al script de cambio de escena por eliminación
30
31    void Start()
32    {
33        animator = GetComponent<Animator>();
34        rb = GetComponent<Rigidbody2D>();
35
36        if (boxCollider == null)
37        {
38            boxCollider = GetComponent<BoxCollider2D>();
39        }
40    }
}

```

Figura 6. Códigos de programación: movimiento de Amaru.

Como se puede evidenciar en las partes previas, el videojuego tiene contenido lúdico y educativo. Se crearon elementos visuales, buscándolos simplificarlos sin que estos pierdan su esencia, característica particular de todo diseño elaborado en estilo pixel art. De igual manera se destacó la importancia de fortalecer nuestra cultura mediante todas las nuevas herramientas al alcance, y poder adaptar y enseñar más sobre lo nuestro sabiendo aprovechar la tecnología que tenemos hoy en día. Ver anexo 5 (elaboración y jugabilidad).

Finalmente, se subió en archivo ZIP del videojuego para que esté disponible para los usuarios. Se usó la plataforma Itch.io, plataforma versátil para la publicación de videojuegos independiente, ya sea para vender o descargar; como medio para hacer público el proyecto. Ver anexos link de acceso.



*Figura 7.* Vista lateral en la elaboración del videojuego.



*Figura 8.* Vista de cámara del videojuego.

## CRONOGRAMA

En este apartado podemos observar la organización que se llevó a cabo para la elaboración del videojuego 2D ambientado en paisajes y mitología Inca en el Ecuador. Se elaboró un cronograma que ayude en dividir cada uno de los objetivos específicos.

### Cronograma de Actividades

Proyecto: Videojuego 2D ambientado en paisajes y mitología Inca en el Ecuador

Objetivos/Actividades	TIEMPO																								
	MES 1				MES 2				MES 3				MES 4				MES 5				MES 6				
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
<b>1</b> Objetivo específico # 1 Investigar las características geográficas, culturales y naturales de varios paisajes del Ecuador que serán representados en el videojuego.																									
1.1 Análisis de fuentes bibliográficas y documentales																									
1.2 Analizar la cultura y mitología Inca en el Ecuador para integrarlas en el diseño del juego																									
1.3 Recopilación de material audiovisual																									
<b>Entregable # 1</b>																									
<b>2</b> Objetivo específico # 2 Analizar las mitología y cultura Inca ene el Ecuador para identificar los elementos clave que serán incorporados en la narrativa y diseño del videojuego 2D																									
2.1 Selección de dios a incorporar en el proyecto																									
2.3 Selección de los lugares turísticos para el desarrollo de los escenarios																									
2.4 Elaboración de la narrativa del videojuego																									
<b>Entregable # 2</b>																									
<b>3</b> Objetivo específico # 3 Desarrollar un videojuego 2D ambientado en los diversos paisajes del Ecuador ofreciendo una experiencia visual y lúdica que destaque la riqueza natural y cultural del país.																									
3.1 Desarrollo de personajes basados en lo investigado																									
3.2 Desarrollo de paisajes para los escenarios de los niveles																									
3.3 Implementación de elementos visuales en Unity																									
3.4 Pruebas y ajustes de diseño																									
<b>Entregable # 3</b>																									
Informe final																									

## PRESUPUESTO

**Proyecto:** Videojuego 2D ambientado en paisajes y mitología Inca en el Ecuador  
**Alumnos:** Fernanda Daniela Cobos López  
Jean Carlos Pilay Rios

RUBROS		Total
<b>REMUNERACIONES</b>		
Mano de obra (horas trabajadas en la elaboración del videojuego)		\$ 650,00
<b>VIÁTICOS Y PASAJES</b>		
Transporte para investigación (viajes al interior)		\$ 100,00
<b>GASTOS EN INFORMÁTICA</b>		
Licencias de software (Unity, Photoshop, etc.)		\$ 65,00
Arrendamientos de Equipos Informáticos		\$ 30,00
<b>SUMINISTROS Y MATERIALES</b>		
Alimentos y bebidas		\$ 50,00
Materiales de oficina (papelería, bolígrafos, etc.)		\$ 20,00
Materiales de computación (memorias USB, discos duros, etc.)		\$ 100,00
<b>TOTAL</b>		\$ 1.015,00

## CONCLUSIONES

El desarrollo del videojuego 2D, titulado Amaru y la espada Kawsaykuna, ambientado en paisajes y mitología ecuatoriana, se presenta como una herramienta para preservar y difundir la riqueza cultural del país.

Para el primer objetivo específico, se llevó a cabo una investigación sobre las características geográficas, culturales y naturales de los paisajes del Ecuador. Esta recopilación permitió escoger los escenarios que serán representados mediante el estilo píxel art en los niveles del videojuego. Todo se recopiló mediante fuentes bibliográficas e imágenes para obtener información de los lugares turísticos en el país, y poder así, obtener diseños donde destaquen su elemento principal y se respete su composición y riqueza única.

Seguidamente, para el segundo objetivo, se escogieron los lugares donde más se asentaron los Inca en Ecuador. Estos fueron: la Mitad del Mundo, la Laguna de Quilotoa y las Ruinas de Ingapirca. Dos de estos, la Mitad del Mundo y la Laguna de Quilotoa; fueron lugares donde se establecieron los Incas por la variedad y facilidad de recursos que obtenían. Por otro lado, la Laguna de Quilotoa es un lugar que se destaca por su riqueza y belleza natural propia.

Además, gracias a la recopilación de información, se escogió al Dios Inti como enemigo principal. Él era el Dios del Sol, un Dios muy respetado y honrado por los Incas; también lo simbolizaban como vida, prosperidad y tiempo.

En cuanto al tercer objetivo específico, el videojuego ofrece una experiencia lúdica destacando las riquezas naturales del Ecuador. Esto no solo fue elaborado como herramienta de entretenimiento, sino, también muestra en cada nivel cierta información concisa de los lugares que los jugadores van conociendo a medida que va avanzando en el juego.

Finalmente, en relación con el objetivo general. El videojuego cumple con su propósito de promover y difundir conocimiento de la cultura Inca en el Ecuador. A través de

la interactividad, la narrativa y el arte. Se elaboró una propuesta que revaloriza el patrimonio cultural y lo presenta en un formato accesible, y fomenta el conocimiento de lugares únicos del país.

## REFERENCIAS BIBLIOGRÁFICAS

### Referencias

Almeida, L., & Franco, A. (2011). Critical thinking: Its relevance for education in a shifting society. *Revista de Psicología*, págs. 175-195.

Alvarracin, C. S. (2015). *Diseño de un libro ilustrado 3d para aportar a la conservación de la mitología ecuatorina en adolescentes*. Cuenca.

Américo, M., Grande, F. C., & Silva, J. F. (2015). *Un acercamiento al eduentretenimiento*. Recuperado el 28 de 7 de 2024, de [http://sedici.unlp.edu.ar/bitstream/handle/10915/45021/documento\\_completo.pdf?squence=1](http://sedici.unlp.edu.ar/bitstream/handle/10915/45021/documento_completo.pdf?squence=1)

Argudo González, K. (17 de Febrero de 2024). *Expreso*. Obtenido de Expreso: <https://www.expreso.ec/ciencia-y-tecnologia/kubox-abre-puertas-primera-arena-esports-ecuador-190016.html>

Carranza, A. (22 de Mayo de 2022). *crehana*. Obtenido de <https://www.crehana.com/blog/estilo-vida/programas-para-hacer-pixel-art/>

Classlife. (7 de Febrero de 2023). Obtenido de Edutainment: qué es y qué beneficios tiene: <https://www.classlife.education/blog/que-es-edutainment/>

culturaypatrimonio. (11 de mazo de 2015). Obtenido de culturaypatrimonio: <https://www.culturaypatrimonio.gob.ec/el-aya-uma-ayuda-a-mantener-el-orden-cosmico/>

- El hotele cuatresures quito. (4 de Abril de 2024). Obtenido de hotelecuatresuresquito:  
<https://hotelecuatresuresquito.ec/es/festivales-para-celebrar-la-cultura-ecuatoriana-que-no-te-puedes-perder/>
- exoticca. (s.f.). Obtenido de exoticca: <https://www.exoticca.com/es/america/america-del-sur/ecuador/eventos>
- Guerra, J., & Revuelta, F. I. (2015). *Videojuegos precursores de emociones positivas: propuesta metodológica con Minecraft en el aula hospitalaria*. IJERI: International Journal of Educational Research and Innovation.
- Instituto de Fomento a la Creatividad y la Innovac.* (02 de Mayo de 2018). Obtenido de Instituto de Fomento a la Creatividad y la Innovac:  
<https://creatividad.gob.ec/2018/05/02/to-leave-videojuego-ecuatoriano-de-calidad-mundial/>
- Instituto de Fomento a la Creatividad y la Innovación. (14 de Marzo de 2023). *Cultura y patrimonio*. Obtenido de Cultura y patrimonio:  
<https://www.culturaypatrimonio.gob.ec/concursos-publicos-de-fomento-para-gestores-culturales-y-artistas/>
- Jiménez Cajamarca, A. A., & Vásquez Badillo, M. A. (2019). *Espol edu*. Obtenido de Espol edu: <https://www.dspace.espol.edu.ec/handle/123456789/53878>
- Jiménez, J. (24 de Septiembre de 2022). *Observatorio Uartes*. Obtenido de Observatorio Uartes: <https://observatorio.uartes.edu.ec/2022/09/24/pueden-los-videojuegos-impulsar-las-industrias-creativas-en-el-ecuador/#:~:text=En%20el%20Ecuador%20ha%20habido,virtual%20de%20Playstation%20y%20Steam.>
- Kent, S. L. (2001). *The Ultimate History of Video Games: From Pong to Pokémon and Beyond*. Roseville: CA: Prima Publishing.
- Lauren. (2024). Obtenido de Pixel Art zone: <https://pixelartzone.com/pixel-art-en-la-educacion/>

León-Atienza, J. D., García-Herrera, D. G., Cabrera-Berrezueta, L. B., & Erazo-Álvarez, J. C. (15 de Noviembre de 2020). *Videojuegos y enseñanza-aprendizaje de Historia: Una revisión sistemática para la vinculación al currículo ecuatoriano*.

Obtenido de CIENCIAMATRIA:

<https://ojs.cienciamatriarevista.org.ve/index.php/cm/article/view/410/551>

Marín Díaz , V., & Fernández , M. D. (2005). Los videojuegos y su capacidad didácticoformativa. *Pixel-Bit: Revista de medios y educación*.

mitosydioses. (s.f.). Obtenido de mitosydioses: <https://www.mitosydioses.com/supay/>

Moreno Cantano, A. C. (2020). EL VIDEOJUEGO COMO ESPEJO DE LA SOCIEDAD CONTEMPORÁNEA. En A. Moreno, *EL VIDEOJUEGO COMO ESPEJO DE LA SOCIEDAD CONTEMPORÁNEA* (págs. 1-8). Madrid: Barataria.

Mythical Encyclopedia. (s.f.). Obtenido de Mythical Encyclopedia:

<https://mythicalencyclopedia.com/inca-gods-and-goddesses/>

Old World Gods. (s.f.). Obtenido de Old World Gods:

<https://oldworldgods.com/other/quechua-mythology-gods-and-goddesses/>

patrimoniocultura. (s.f.). Obtenido de <https://www.patrimoniocultural.gob.ec/el-pasillo-ecuatoriano/>

patrimoniocultural. (13 de enero de 2003). Obtenido de

<https://www.patrimoniocultural.gob.ec/la-marimba-y-sus-constituyentes/>

Ponce, S. A. (2017). *Google académico*. Obtenido de Google académico:

<file:///C:/Users/user/Downloads/UDLA-EC-TDGI-2017-24.pdf>

Queiroga, A. (2007). *AQB*. Obtenido de <http://www.aqb.com.br/aqb/artigo.php?no=02>

Sedeño Vandellós, A. (2010). Videojuegos como dispositivos culturales: las competencias espaciales en educación. En A. Sedeño Vandellós, *Videojuegos como dispositivos culturales: las competencias espaciales en educación*. (págs. 183-189). Comunicar.

shirani. (octubre de 2017). Obtenido de steemit: <https://steemit.com/art/@shirani/que-es-pixel-art#:~:text=Existen%20dos%20tipos%20de%20Pixel,3D%20en%20un%20espacio%202D.>

U-tad. (3 de octubre de 2023). Obtenido de Centro Universitario de Tecnología y Arte Digital: <https://u-tad.com/pixel-art/>

Valderrama-Ramo, J. A. (2012). *Sinectica*. Obtenido de <https://sinectica.iteso.mx/index.php/SINECTICA/issue/view/42>

## APÉNDICE Y/O ANEXOS

Adjunto link de acceso para descargar ZIP del videojuego:  
<https://ferda-29.itch.io/amaru-ya-la-espada>

### Anexo 1.- (Recopilación de material visual).

Ruinas de Ingapirca



## Laguna de Quilotoa



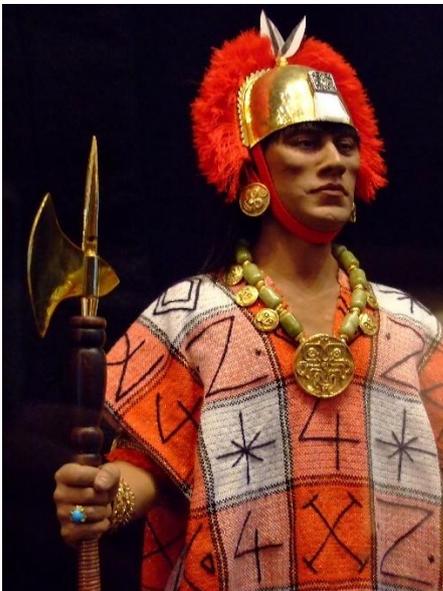
## Mitad del Mundo



## Dios Inti



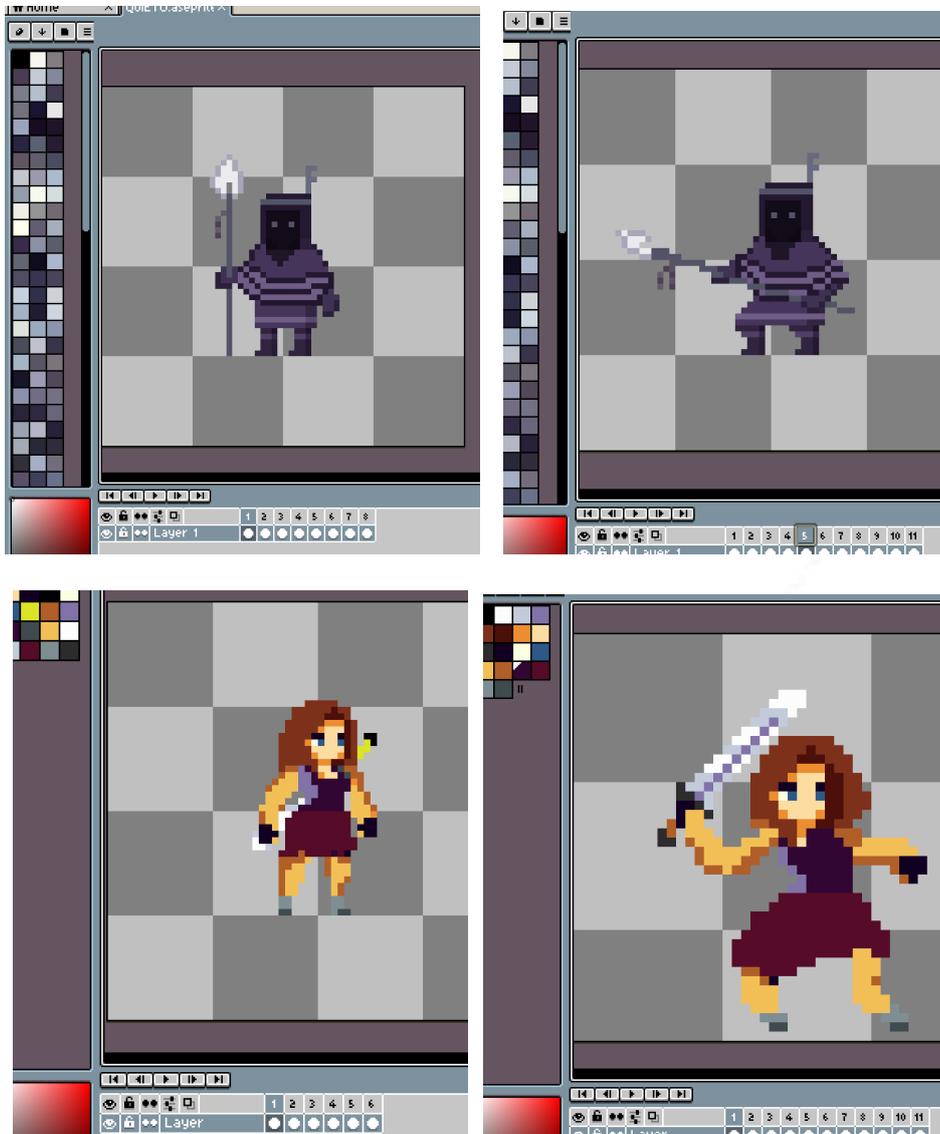
## Personas pertenecientes al ejercito Inca

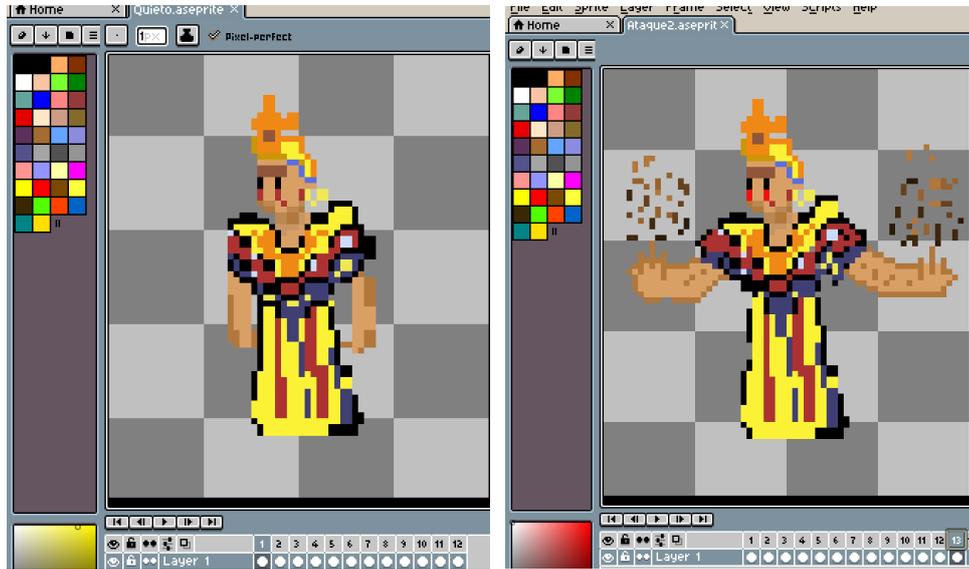


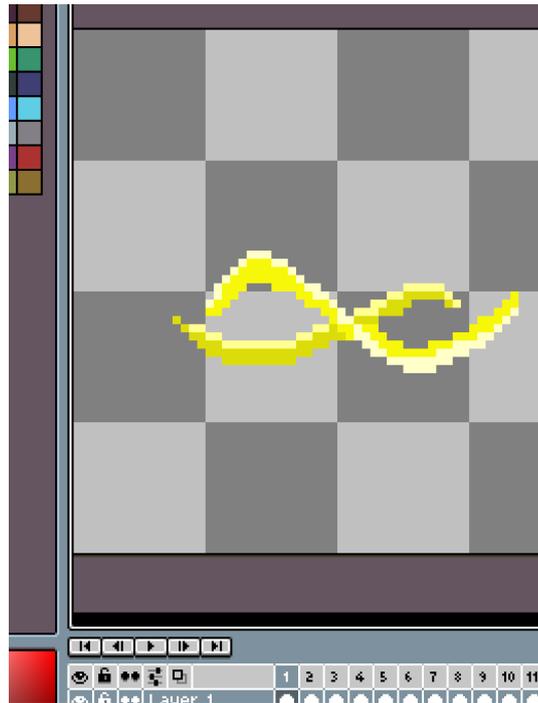
**Anexo 2.- (Selección de lugares, Dios y creación de narrativa).**

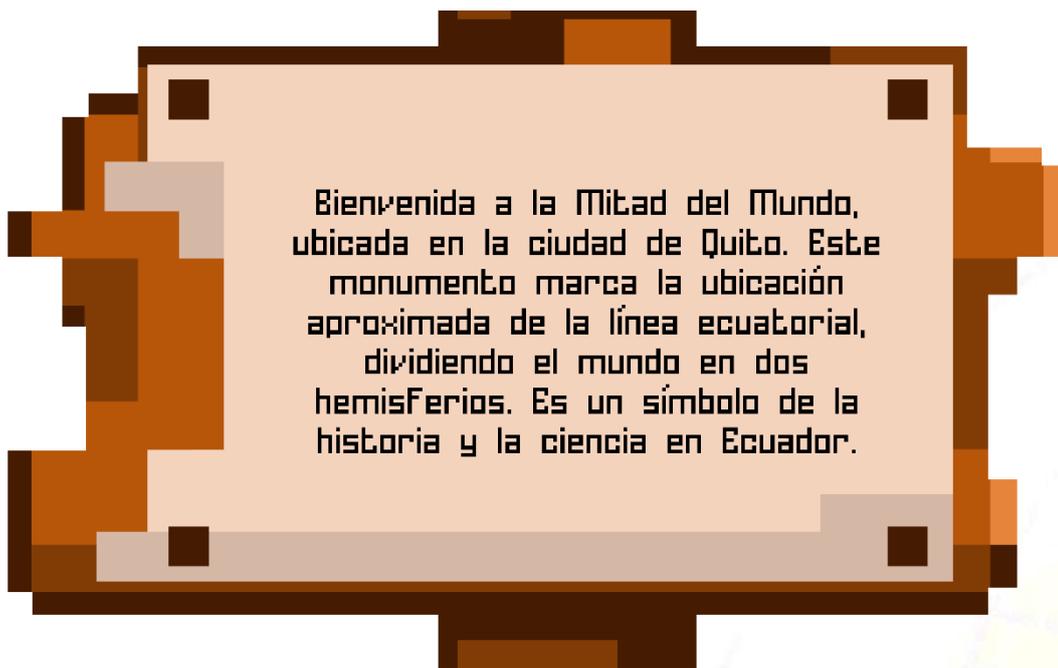
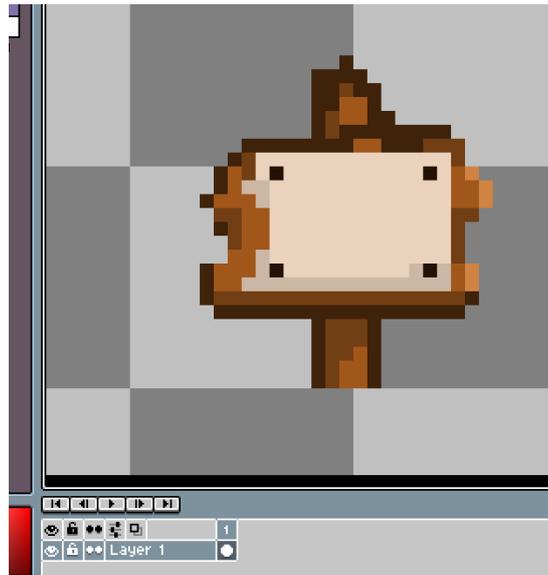


**Anexo 3.- (Diseño y elaboración de animaciones de los personajes, ataques, elementos y escenarios).**







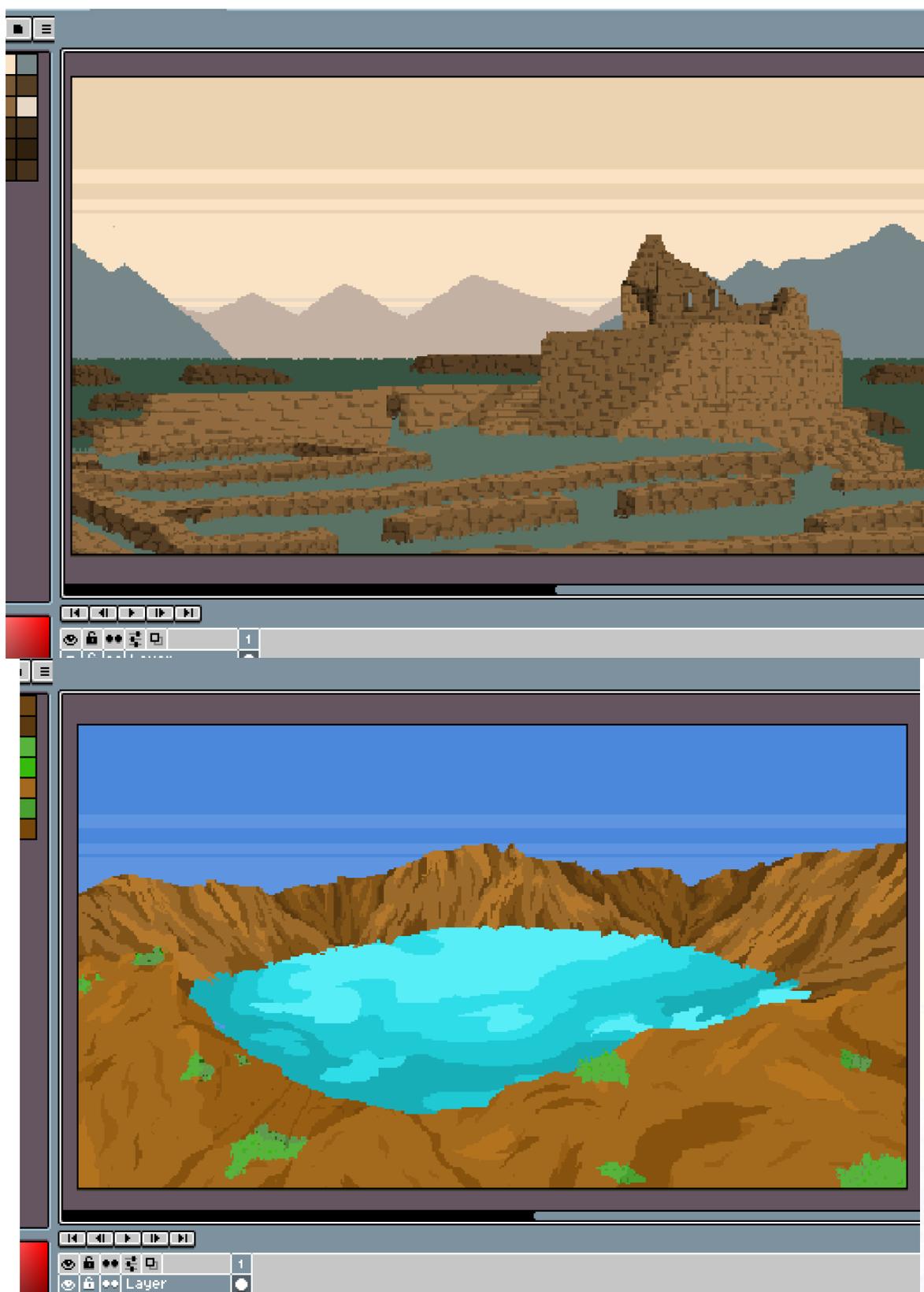


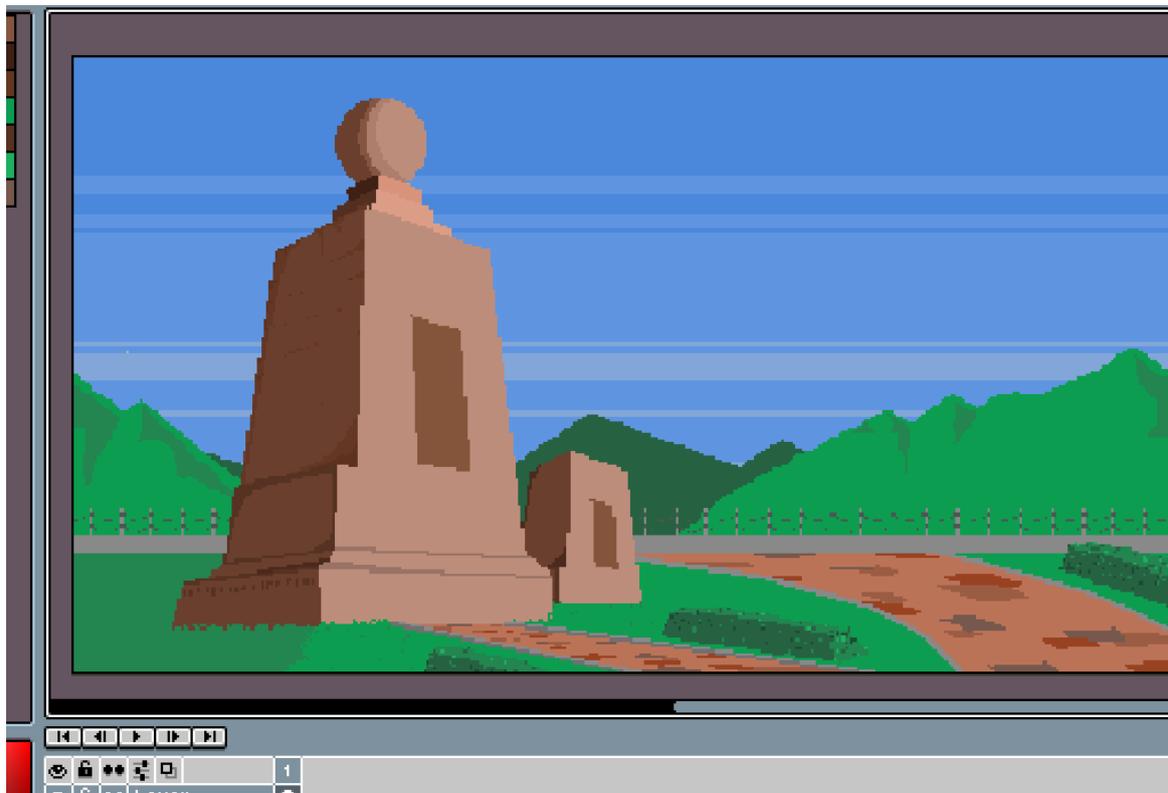
Estás en la Laguna de Quilotoa, una majestuosa caldera volcánica situada en los Andes ecuatorianos. Su laguna, de aguas turquesas, se formó tras la erupción del volcán hace más de 800 años. Es un lugar sagrado para las comunidades indígenas que lo rodean.

Bienvenida a Ingapirca, las ruinas incaicas más importantes de Ecuador. Este sitio, ubicado en la provincia de Cañar, fue un centro ceremonial y observatorio solar del Imperio Inca, donde se fusionaron tradiciones incas y cañaris. Su estructura más emblemática es el Templo del Sol.

Dios Inti: Conocido como el dios del sol, era una de las deidades más veneradas en la cultura Inca. Inti simbolizaba la vida, la prosperidad y el tiempo, siendo esencial para la agricultura y la espiritualidad de su pueblo.







## Anexos 4.- (Programación e integración de elementos)

### Control de personaje (Amaru)

```
Assets > scripts > ControlPersonaje.cs
1 using UnityEngine;
2 using TMPro;
3
4 public class ControlPersonaje : MonoBehaviour
5 {
6     public float velocidad = 5f; // Velocidad de movimiento
7     public float fuerzaSalto = 10f; // Fuerza del salto
8     public float distanciaSuelo = 0.1f; // Distancia para detectar el suelo
9     public LayerMask capaSuelo; // Capa del suelo para el Raycast
10    public Vector3 posicionInicio; // Posición inicial del personaje
11    public ControlVidas controlVidas; // Referencia al ControlVidas para manejar las oportunidades
12
13    public BoxCollider2D boxCollider; // Referencia al BoxCollider2D del personaje
14    public Vector2 tamañoNormal = new Vector2(1f, 2f); // Tamaño del collider cuando está de pie
15    public Vector2 tamañoAgachado = new Vector2(1f, 1f); // Tamaño del collider cuando está agachado
16    public Vector2 offsetNormal = new Vector2(0f, 0f); // Offset del collider cuando está de pie
17    public Vector2 offsetAgachado = new Vector2(0f, -0.5f); // Offset del collider cuando está agachado
18
19    private Animator animator; // Referencia al Animator
20    private Rigidbody2D rb; // Referencia al Rigidbody2D
21    private bool enSuelo = true; // Verifica si está en el suelo
22    private bool estaSaltando = false; // Verifica si está saltando
23    private bool estaAgachado = false; // Verifica si está agachado
24    private bool enAnimacionDeDaño = false; // Indica si está en la animación de daño
25    private bool movimientoBloqueado = false; // Indica si el movimiento está bloqueado
26    private bool entradasBloqueadas = false; // Bloquea todas las entradas, incluidos clics y teclas
27    public int vidaActual = 3; // Vida actual del personaje
28
29    public CambioEscenaPorEliminacion cambioEscena; // Referencia al script de cambio de escena por eliminación
30
31    void Start()
32    {
33        animator = GetComponent<Animator>();
34        rb = GetComponent<Rigidbody2D>();
35
36        if (boxCollider == null)
37        {
38            boxCollider = GetComponent<BoxCollider2D>();
39        }
40    }
41}
```

```
animator = GetComponent<Animator>();
rb = GetComponent<Rigidbody2D>();

if (boxCollider == null)
{
    boxCollider = GetComponent<BoxCollider2D>();
}

posicionInicio = transform.position;

void Update()
{
    // Si las entradas están bloqueadas, no hacer nada
    if (entradasBloqueadas || Time.timeScale == 0 || movimientoBloqueado || enAnimacionDeDaño)
    {
        return;
    }

    enSuelo = Physics2D.Raycast(transform.position, Vector2.down, distanciaSuelo, capaSuelo);

    if (rb.linearVelocity.y > 0 && !enSuelo)
    {
        estaSaltando = true;
        animator.SetBool("isJumping", true);
        animator.SetBool("isFalling", false);
    }
    else if (rb.linearVelocity.y < 0 && !enSuelo)
    {
        estaSaltando = true;
        animator.SetBool("isJumping", false);
        animator.SetBool("isFalling", true);
    }
    else if (enSuelo)
    {
        estaSaltando = false;
        animator.SetBool("isJumping", false);
        animator.SetBool("isFalling", false);
    }
}
```

```
84     }
85     else if (movimiento < 0)
86     {
87         transform.localScale = new Vector3(-1, 1, 1);
88     }
89 }
90
91 if (Input.GetKeyDown(KeyCode.S) && enSuelo)
92 {
93     estaAgachado = true;
94     animator.SetBool("isCrouching", true);
95     CambiarCollider(tamañoAgachado, offsetAgachado);
96 }
97 else if (Input.GetKeyUp(KeyCode.S))
98 {
99     estaAgachado = false;
100    animator.SetBool("isCrouching", false);
101    CambiarCollider(tamañoNormal, offsetNormal);
102 }
103
104 if (Input.GetButtonDown("Jump") && enSuelo && !estaAgachado)
105 {
106    rb.AddForce(Vector2.up * fuerzaSalto, ForceMode2D.Impulse);
107    animator.SetBool("isJumping", true);
108 }
109 }
110
111 void CambiarCollider(Vector2 nuevoTamaño, Vector2 nuevoOffset)
112 {
113     if (boxCollider != null)
114     {
115         boxCollider.size = nuevoTamaño;
116         boxCollider.offset = nuevoOffset;
117     }
118 }
119
```

```
121 {
122     if (!enAnimacionDeDaño)
123     {
124         Debug.Log("Recibiendo daño.");
125         vidaActual -= cantidad;
126
127         enAnimacionDeDaño = true; // Bloquear otras acciones
128         if (animator != null)
129         {
130             animator.SetTrigger("RecibirDaño");
131         }
132
133         // Restauran el control después de 1 segundo (ajusta este tiempo según tu animación)
134         Invoke(nameof(FinDeDaño), 0.50f);
135
136         if (vidaActual <= 0)
137         {
138             Morir();
139         }
140     }
141 }
142
143 private void Morir()
144 {
145     Debug.Log("El jugador ha muerto.");
146 }
147
148 public void FinDeDaño()
149 {
150     enAnimacionDeDaño = false; // Permitir otras acciones nuevamente
151     Debug.Log("Fin de la animación de daño. Control restaurado.");
152 }
153
154 public void MorirPorCaída()
155 {
156     Debug.Log("El personaje cayó al vacío.");
157
158     if (controlVidas != null)
```

```
Assets > scripts > ControlPersonaje.cs
5
{
155
{
156     Debug.Log("El personaje cayó al vacío.");
157
158     if (controlVidas != null)
159     {
160         controlVidas.PenderOportunidad();
161     }
162
163     transform.position = posicionInicio;
164     rb.linearVelocity = Vector2.zero;
165 }
166
167 public void ActualizarPosicionInicio(Vector3 nuevaPosicion)
168 {
169     posicionInicio = nuevaPosicion;
170     Debug.Log("Posición de inicio actualizada a: " + posicionInicio);
171 }
172
173 // Método para bloquear o desbloquear el movimiento
174 public void BloquearMovimiento(bool bloquear)
175 {
176     movimientoBloqueado = bloquear;
177     Debug.Log($"Movimiento bloqueado: {bloquear}");
178 }
179
180 // Método para bloquear o desbloquear todas las entradas
181 public void BloquearEntradas(bool bloquear)
182 {
183     entradasBloqueadas = bloquear;
184     Debug.Log($"Entradas bloqueadas: {bloquear}");
185 }
186
187 // Método para verificar si las entradas están bloqueadas
188 public bool EntradasBloqueadas()
189 {
190     return entradasBloqueadas;
191 }
192
192
```

```
192
193 // Método para verificar si el personaje está en el suelo
194 public bool EstaEnSuelo()
195 {
196     return enSuelo;
197 }
198
199 // Método para ejecutar un ataque especial
200 public void EjecutarAtaqueEspecial(int contadorGolpes, int golpesParaEspecial)
201 {
202     if (contadorGolpes >= golpesParaEspecial)
203     {
204         animator?.SetTrigger("Ataque2");
205         Debug.Log("Animación 'Ataque2' activada.");
206     }
207     else
208     {
209         Debug.Log("No se cumplen las condiciones para el ataque especial.");
210     }
211 }
212
213 void OnTriggerEnter2D(Collider2D collision)
214 {
215     if (collision.CompareTag("TriggerCambioEscena") && cambioEscena != null)
216     {
217         Debug.Log("Jugador tocó el trigger de cambio de escena.");
218         if (GameObject.FindGameObjectsWithTag("Enemigo").Length == 0) // Verificar si no hay enemigos
219         {
220             cambioEscena.CambiarEscena();
221         }
222         else
223         {
224             Debug.Log("Aún hay enemigos en la escena. No se puede cambiar.");
225         }
226     }
227 }
228
229
```

Ln 2, Col 13 Spaces:4 UTF-8 CRLF

## Control de vida

```
1 using UnityEngine;
2 using TMPro; // Para trabajar con TextMeshPro
3
4 public class ControlVidas : MonoBehaviour
5 {
6     public TextMeshProUGUI vidaTexto; // Referencia al texto en el canvas
7     public GameManager gameOverManager; // Referencia al script GameManager
8     private int oportunidadesRestantes = 3; // Número inicial de oportunidades
9
10    void Start()
11    {
12        ActualizarTexto(); // Inicializar el texto con las oportunidades restantes
13    }
14
15    public void PerderOportunidad()
16    {
17        if (oportunidadesRestantes > 0)
18        {
19            oportunidadesRestantes--; // Reducir las oportunidades
20            ActualizarTexto(); // Actualizar el texto en pantalla
21        }
22
23        if (oportunidadesRestantes <= 0)
24        {
25            // Llamar al GameManager para mostrar el panel de Game Over
26            if (gameOverManager != null)
27            {
28                gameOverManager.MostrarGameOver();
29            }
30            else
31            {
32                Debug.LogWarning("No se asignó un GameManager en ControlVidas.");
33            }
34        }
35    }
36
37    private void ActualizarTexto()
38    {
39        if (vidaTexto != null)
```

```
25        // Llamar al GameManager para mostrar el panel de Game Over
26        if (gameOverManager != null)
27        {
28            gameOverManager.MostrarGameOver();
29        }
30        else
31        {
32            Debug.LogWarning("No se asignó un GameManager en ControlVidas.");
33        }
34    }
35 }
36
37 private void ActualizarTexto()
38 {
39     if (vidaTexto != null)
40     {
41         vidaTexto.text = oportunidadesRestantes.ToString();
42     }
43     else
44     {
45         Debug.LogWarning("No se asignó un TextMeshProUGUI al texto de oportunidades.");
46     }
47 }
48 }
49 }
```

## Ataque de personaje (Amaru)

```
108 // Detectar todos los colliders en el rango
109 Collider2D[] collidersDetectados = Physics2D.OverlapCircleAll(transform.position, rangoAtaque);
110
111 foreach (Collider2D collider in collidersDetectados)
112 {
113     // Detectar enemigos con barra de vida
114     BarraDeVidaEnemigo barraVida = collider.GetComponentInChildren<BarraDeVidaEnemigo>();
115     if (barraVida != null)
116     {
117         barraVida.RecibirDaño(dañoAtaqueEspecial);
118         Debug.Log($"Ataque especial infligido a enemigo: {dañoAtaqueEspecial}");
119     }
120
121     // Detectar si el objeto es un jefe
122     JefeBase jefe = collider.GetComponent<JefeBase>();
123     if (jefe != null)
124     {
125         jefe.RecibirDaño(dañoAtaqueEspecial);
126         Debug.Log($"Jefe recibió daño especial: {dañoAtaqueEspecial}");
127     }
128 }
129
130 Invoke(nameof(FinDeAtaque), cooldownEspecial); // Desbloquear movimiento después del cooldown
131
132 void FinDeAtaque()
133 {
134     controlPersonaje.BloquearMovimiento(false); // Permitir movimiento nuevamente
135 }
136
137 void OnDrawGizmosSelected()
138 {
139     Gizmos.color = Color.red;
140     Gizmos.DrawWireSphere(transform.position, rangoAtaque);
141 }
142
143 }
144
145
146 Debug.Log($"Golpe básico infligido a enemigo: {dañoAtaqueBasico}");
147 }
148
149 // Detectar si el objeto es un jefe
150 JefeBase jefe = collider.GetComponent<JefeBase>();
151 if (jefe != null)
152 {
153     jefe.RecibirDaño(dañoAtaqueBasico);
154     Debug.Log($"Jefe recibió daño básico: {dañoAtaqueBasico}");
155     contadorGolpes++;
156
157     if (contadorGolpes >= golpesParaEspecial)
158     {
159         Debug.Log("Ataque especial desbloqueado.");
160     }
161 }
162 }
163
164 Invoke(nameof(FinDeAtaque), cooldownBasico); // Desbloquear movimiento después del cooldown
165 }
166
167 void AtacarEspecial()
168 {
169     tiempoUltimoEspecial = Time.time;
170     contadorGolpes = 0; // Reiniciar el contador de golpes
171
172     if (animator != null)
173     {
174         animator.SetTrigger("Ataque2");
175     }
176
177     controlPersonaje.BloquearMovimiento(true); // Bloquear movimiento
178
179     // Detectar todos los colliders en el rango
180 Collider2D[] collidersDetectados = Physics2D.OverlapCircleAll(transform.position, rangoAtaque);
181 }
```

```

40     {
41         return;
42     }
43
44     if (Input.GetMouseButtonDown(0) && Time.time >= tiempoUltimoBasico + cooldownBasico) // Ataque básico
45     {
46         AtacarBasico();
47     }
48
49     if (contadorGolpes >= golpesParaEspecial && Input.GetMouseButtonDown(1) && Time.time >= tiempoUltimoEspecial + cooldownEspecial) // At
50     {
51         AtacarEspecial();
52     }
53 }
54
55 void AtacarBasico()
56 {
57     tiempoUltimoBasico = Time.time;
58
59     if (animator != null)
60     {
61         animator.SetTrigger("Ataque");
62     }
63
64     controlPersonaje.BloquearMovimiento(true); // Bloquear movimiento
65
66     // Detectar todos los colliders en el rango
67     Collider2D[] collidersDetectados = Physics2D.OverlapCircleAll(transform.position, rangoAtaque);
68
69     foreach (Collider2D collider in collidersDetectados)
70     {
71         // Detectar enemigos con barra de vida
72         BarraDeVidaEnemigo barraVida = collider.GetComponentInChildren<BarraDeVidaEnemigo>();
73         if (barraVida != null)
74         {
75
76         }
77     }
78 }
79
80 using UnityEngine;
81
82 public class AtaquesPersonaje : MonoBehaviour
83 {
84     [Header("Configuración de Ataques")]
85     public int dañoAtaqueBasico = 10;
86     public int dañoAtaqueEspecial = 30;
87     public int golpesParaEspecial = 5;
88     public float rangoAtaque = 1.5f;
89     public float cooldownBasico = 0.5f;
90     public float cooldownEspecial = 5f;
91
92     private Animator animator;
93     private int contadorGolpes = 0;
94     private float tiempoUltimoBasico = -Mathf.Infinity;
95     private float tiempoUltimoEspecial = -Mathf.Infinity;
96     private ControlPersonaje controlPersonaje;
97
98     void Start()
99     {
100         animator = GetComponent<Animator>();
101         controlPersonaje = GetComponent<ControlPersonaje>();
102
103         if (controlPersonaje == null)
104         {
105             Debug.LogError("ControlPersonaje no está asignado o no está en el mismo GameObject.");
106         }
107     }
108
109     void Update()
110     {
111         // Bloquear ataques si las entradas están deshabilitadas
112         if (controlPersonaje != null && controlPersonaje.EntradasBloqueadas())
113         {
114             return;
115         }
116
117         // Verificar si el personaje está en el suelo antes de permitir los ataques
118         if (controlPersonaje != null && !controlPersonaje.EstaEnSuelo())

```

## Scripts del enemigo genérico

```
95 void IniciarDash()
96 {
97     enDash = true;
98     StartCoroutine(EjecutarDash());
99 }
100
101 private System.Collections.IEnumerator EjecutarDash()
102 {
103     float tiempoInicio = Time.time;
104
105     while (Time.time < tiempoInicio + tiempoDash)
106     {
107         // Solo aplicar movimiento horizontal durante el dash
108         rb.linearVelocity = new Vector2(direccionDash.x * velocidadDash, rb.linearVelocity.y);
109         yield return null;
110     }
111
112     // Detener el movimiento horizontal al finalizar el dash, pero mantener la caída
113     rb.linearVelocity = new Vector2(0, rb.linearVelocity.y);
114     enDash = false; // Permitir otro dash
115     tiempoUltimoDash = Time.time; // Registrar el tiempo del último dash
116 }
117
118 void OnDrawGizmosSelected()
119 {
120     // Dibujar el rango de detección en el editor
121     Gizmos.color = Color.red;
122     Gizmos.DrawWireSphere(transform.position, rangoDeteccion);
123 }
124 }
125
126
127
```

```
77     {
78         transform.localScale = new Vector3(Mathf.Abs(escalaInicial), transform.localScale.y, transform.localScale.z);
79     }
80 }
81
82 void ActivarAnimacionVDash()
83 {
84     if (animator != null)
85     {
86         // Activar la animación de dash
87         animator.SetTrigger("Dash");
88         Debug.Log("Animación Dash activada.");
89     }
90
91     // Esperar el tiempo configurado antes de realizar el dash
92     Invoke(nameof(IniciarDash), tiempoEsperaAnimacion);
93 }
94
95 void IniciarDash()
96 {
97     enDash = true;
98     StartCoroutine(EjecutarDash());
99 }
100
101 private System.Collections.IEnumerator EjecutarDash()
102 {
103     float tiempoInicio = Time.time;
104
105     while (Time.time < tiempoInicio + tiempoDash)
106     {
107         // Solo aplicar movimiento horizontal durante el dash
108         rb.linearVelocity = new Vector2(direccionDash.x * velocidadDash, rb.linearVelocity.y);
109         yield return null;
110     }
111
112     // Detener el movimiento horizontal al finalizar el dash, pero mantener la caída
113     rb.linearVelocity = new Vector2(0, rb.linearVelocity.y);
114 }
```

```

40
41 void Update()
42 {
43     // Detectar si el jugador está en el rango
44     Collider2D jugadorDetectado = Physics2D.OverlapCircle(transform.position, rangoDeteccion, capaJugador);
45
46     if (jugadorDetectado != null && !enDash && Time.time >= tiempoUltimoDash + tiempoEsperaEntreDashes)
47     {
48         jugador = jugadorDetectado.transform;
49         CalcularDireccionDash();
50         ActivarAnimacionVDash();
51     }
52 }
53
54 void CalcularDireccionDash()
55 {
56     if (jugador != null)
57     {
58         // Calcular la dirección hacia el jugador en el eje X (derecha o izquierda)
59         direccionDash = new Vector2(Mathf.Sign(jugador.position.x - transform.position.x), 0);
60
61         // Girar hacia la dirección calculada
62         GirarHaciaDireccionInversa(direccionDash.x);
63
64         Debug.Log("Dirección del dash calculada: " + direccionDash);
65     }
66 }
67
68 void GirarHaciaDireccionInversa(float direccionX)
69 {
70     // Si el dash es hacia la derecha, el enemigo debe mirar a la izquierda
71     if (direccionX > 0)
72     {
73         transform.localScale = new Vector3(-Mathf.Abs(escalaInicial), transform.localScale.y, transform.localScale.z);
74     }
75     // Si el dash es hacia la izquierda, el enemigo debe mirar a la derecha
76     else if (direccionX < 0)
77     {

```

```

1 using UnityEngine;
2
3 public class EnemigoDash : MonoBehaviour
4 {
5     [Header("Configuración del Dash")]
6     public float velocidadDash = 10f; // Velocidad del dash
7     public float tiempoDash = 0.2f; // Duración del dash
8     public float tiempoEsperaAnimacion = 1f; // Tiempo entre la animación y el inicio del dash
9     public float tiempoEsperaEntreDashes = 2f; // Tiempo de espera entre cada dash
10
11     [Header("Configuración de Detección")]
12     public float rangoDeteccion = 5f; // Rango en el que detecta al jugador
13     public LayerMask capaJugador; // Capa del jugador para detección
14
15     [Header("Componentes")]
16     public Rigidbody2D rb; // Referencia al Rigidbody2D
17     public Animator animator; // Referencia al Animator
18
19     private Transform jugador; // Referencia al jugador detectado
20     private bool enDash = false; // Controla si el enemigo está haciendo un dash
21     private Vector2 direccionDash; // Dirección fija durante el dash
22     private float tiempoUltimoDash = 0f; // Controla el tiempo del último dash
23     private float escalaInicial; // Almacena la escala inicial en X para determinar la dirección
24
25     void Start()
26     {
27         if (rb == null)
28         {
29             rb = GetComponent<Rigidbody2D>();
30         }
31
32         if (animator == null)
33         {
34             animator = GetComponent<Animator>();
35         }
36
37         // Guardar la escala inicial del enemigo
38         escalaInicial = transform.localScale.x;
39     }

```

## Scripts – movimiento del enemigo genérico

```
1 using UnityEngine;
2
3 public class MovimientoEnemigo : MonoBehaviour
4 {
5     public float velocidad = 2f; // Velocidad de movimiento
6     public Transform puntoA; // Primer punto de patrulla
7     public Transform puntoB; // Segundo punto de patrulla
8     private Transform objetivoActual; // Punto al que se dirige actualmente
9     private SpriteRenderer spriteRenderer; // Referencia al SpriteRenderer para voltear el sprite
10
11     void Start()
12     {
13         objetivoActual = puntoA; // Comienza yendo al punto A
14         spriteRenderer = GetComponent<SpriteRenderer>();
15     }
16
17     void Update()
18     {
19         // Moverse hacia el objetivo actual
20         transform.position = Vector2.MoveTowards(transform.position, objetivoActual.position, velocidad * Time.deltaTime);
21
22         // Cambiar el objetivo si llega al punto actual
23         if (Vector2.Distance(transform.position, objetivoActual.position) < 0.1f)
24         {
25             objetivoActual = objetivoActual == puntoA ? puntoB : puntoA;
26             spriteRenderer.flipX = !spriteRenderer.flipX; // Voltear el sprite
27         }
28     }
29 }
30
```

## Scripts del Jefe (Inti)

```
39 }
40
41 private void ActivarMovimiento()
42 {
43     velocidadInicial = velocidadMaxima; // Activar el movimiento con la velocidad máxima
44 }
45
46 private void OnTriggerEnter2D(Collider2D collision)
47 {
48     if (collision.CompareTag("Player"))
49     {
50         // Reducir vida del jugador
51         VidaConTileset vidaConTileset = collision.GetComponent<VidaConTileset>();
52         if (vidaConTileset != null)
53         {
54             vidaConTileset.RecibirDaño(daño); // Aplicar daño al jugador
55         }
56
57         // Activar la animación de daño
58         ControlPersonaje controlPersonaje = collision.GetComponent<ControlPersonaje>();
59         if (controlPersonaje != null)
60         {
61             controlPersonaje.RecibirDaño(daño); // Reutilizar la lógica de daño existente
62         }
63
64         // Destruir el ataque tras impactar
65         Destroy(gameObject);
66     }
67 }
68 }
69
```

```
1 using UnityEngine;
2
3 public class AtaqueEspecial : MonoBehaviour
4 {
5     public float velocidadInicial = 0f; // Velocidad inicial del ataque
6     public float velocidadMaxima = 10f; // Velocidad máxima del ataque
7     public float tiempoRetraso = 1f; // Tiempo antes de empezar a moverse
8     public float crecimientoEscala = 0.1f; // Velocidad de crecimiento
9     public float tiempoDeVida = 5f; // Tiempo antes de que el ataque se destruya
10    public int daño = 2; // Cantidad de daño que causa al jugador
11    private bool movimientoActivado = false; // Controla cuándo comienza a moverse
12    private Vector3 direccionMovimiento; // Dirección del ataque
13
14    public void ConfigurarDireccion(Vector3 direccion)
15    {
16        direccionMovimiento = direccion.normalized; // Guardamos la dirección que recibe
17        // Girar el ataque según la dirección
18        float angulo = Mathf.Atan2(direccionMovimiento.y, direccionMovimiento.x) * Mathf.Rad2Deg;
19        transform.rotation = Quaternion.Euler(0, 0, angulo);
20    }
21
22    private void Start()
23    {
24        // Retrasar el movimiento
25        Invoke(nameof(ActivarMovimiento), tiempoRetraso);
26
27        // Programar destrucción del ataque
28        Destroy(gameObject, tiempoDeVida);
29    }
30
31    private void Update()
32    {
33        // Si el movimiento está activado, mover el ataque y hacerlo crecer
34        if (movimientoActivado)
35        {
36            transform.Translate(direccionMovimiento * velocidadInicial * Time.deltaTime);
37            transform.localScale += Vector3.one * crecimientoEscala * Time.deltaTime;
38        }
39    }
}
```

## Daño del jege (Inti)

```
1 using UnityEngine;
2
3 public class DañoEnemigo : MonoBehaviour
4 {
5     public int cantidadDaño = 1; // Cantidad de vida que quitará este enemigo
6
7     private void OnCollisionEnter2D(Collision2D collision)
8     {
9         // Verifica si el objeto con el que colisiona es el jugador
10        if (collision.gameObject.CompareTag("Player"))
11        {
12            // Obtener el script de vida del jugador
13            VidaConTileset vidaJugador = collision.gameObject.GetComponent<VidaConTileset>();
14            if (vidaJugador != null)
15            {
16                vidaJugador.RecibirDaño(cantidadDaño); // Reducir la vida del jugador
17            }
18        }
19    }
20 }
21
22
```

## Jefe base (Inti)

```
217 private System.Collections.IEnumerator Morir()
218 {
219     Debug.Log("El jefe ha muerto.");
220     puedeAtacar = false; // Deshabilitar ataques
221     velocidadMovimiento = 0f; // Detener el movimiento
222
223     if (animator != null)
224     {
225         animator.SetTrigger("Morir");
226     }
227
228     // Oscurecer la pantalla
229     float tiempo = 0f;
230     while (tiempo < duracionTransicion)
231     {
232         tiempo += Time.deltaTime;
233         pantallaOscura.alpha = Mathf.Lerp(0f, 1f, tiempo / duracionTransicion);
234         yield return null;
235     }
236
237     // Cambiar de escena
238     SceneManager.LoadScene(nombreSiguienteEscena);
239 }
240 }
241
242
243
244 {
245     puedeAtacar = true;
246 }
247
248 private void OnCollisionEnter2D(Collision2D collision)
249 {
250     if (collision.gameObject.CompareTag("Player"))
251     {
252         ControlPersonaje controlPersonaje = collision.gameObject.GetComponent<ControlPersonaje>();
253         if (controlPersonaje != null)
254         {
255             controlPersonaje.RecibirDaño(1);
256         }
257     }
258 }
259
260 // Método para recibir daño
261 public void RecibirDaño(int cantidad)
262 {
263     vidaActual -= cantidad;
264     Debug.Log($"Jefe recibió {cantidad} de daño. Vida restante: {vidaActual}");
265
266     if (animator != null)
267     {
268         animator.SetTrigger("Daño");
269     }
270
271     if (vidaActual <= 0)
272     {
273         StartCoroutine(Morir());
274     }
275 }
276
277 private System.Collections.IEnumerator Morir()
278 {
279     Debug.Log("El jefe ha muerto.");
280     puedeAtacar = false; // Deshabilitar ataques
281     velocidadMovimiento = 0f; // Detener el movimiento
282 }
```

```
148     }
149     }
150
151     private void LanzarRayoDoble()
152     {
153         animator?.SetTrigger("Ataque2");
154         foreach (Transform spawn in puntosDisparoDoble)
155         {
156             GameObject rayo = Instantiate(rayoPrefab, spawn.position, Quaternion.identity);
157             RayoFrontal rayoScript = rayo.GetComponent<RayoFrontal>();
158             if (rayoScript != null)
159             {
160                 Vector3 direccion = jugador.position.x > transform.position.x ? Vector3.left : Vector3.right;
161                 rayoScript.ConfigurarDireccion(direccion);
162             }
163         }
164     }
165
166     private void LanzarAtaqueEspecial()
167     {
168         animator?.SetTrigger("AtaqueEspecial");
169         CrearAtaqueEspecial(Vector3.right);
170         CrearAtaqueEspecial(Vector3.left);
171     }
172
173     private void CrearAtaqueEspecial(Vector3 direccion)
174     {
175         GameObject especial = Instantiate(ataqueEspecialPrefab, puntoDisparo.position, Quaternion.identity);
176         AtaqueEspecial ataqueScript = especial.GetComponent<AtaqueEspecial>();
177         if (ataqueScript != null)
178         {
179             ataqueScript.ConfigurarDireccion(direccion);
180         }
181     }
182
183     public void HabilitarAtaques()
184     {
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

```
75     {
76         transform.localScale = new Vector3(-Mathf.Abs(escalaActual.x), escalaActual.y, escalaActual.z);
77     }
78 }
79
80 private void Move()
81 {
82     if (puntosMovimiento.Length == 0) return;
83
84     Transform destino = puntosMovimiento[puntoActual];
85     transform.position = Vector3.MoveTowards(transform.position, destino.position, velocidadMovimiento * Time.deltaTime);
86     animator.SetBool("EstaMoviendose", true);
87
88     if (Vector3.Distance(transform.position, destino.position) < 0.1f)
89     {
90         puntoActual = (puntoActual + 1) % puntosMovimiento.Length;
91         EstablecerQuieto();
92     }
93 }
94
95 private void EstablecerQuieto()
96 {
97     estaQuieto = true;
98     tiempoQuieto = Random.Range(tiempoQuietoMin, tiempoQuietoMax);
99     animator.SetBool("EstaMoviendose", false);
100 }
101
102 private void ManejarQuieto()
103 {
104     tiempoQuieto -= Time.deltaTime;
105     if (tiempoQuieto <= 0)
106     {
107         estaQuieto = false;
108     }
109 }
110
111
112
113
114 private void Update()
115 {
116     // Si el jefe no puede atacar, no ejecuta su lógica de ataque
117     if (!puedeAtacar) return;
118
119     // Rotar hacia el jugador
120     MirarAlJugador();
121
122     // Manejar movimiento o estado quieto
123     if (estaQuieto)
124     {
125         ManejarQuieto();
126     }
127     else
128     {
129         Move();
130     }
131
132     // Manejar el ataque
133     ManejarAtaque();
134 }
135
136 private void MirarAlJugador()
137 {
138     if (jugador == null) return;
139
140     float direccion = jugador.position.x - transform.position.x;
141     Vector3 escalaActual = transform.localScale;
142
143     if (direccion < 0)
144     {
145         transform.localScale = new Vector3(Mathf.Abs(escalaActual.x), escalaActual.y, escalaActual.z);
146     }
147     else if (direccion > 0)
148     {
149         transform.localScale = new Vector3(-Mathf.Abs(escalaActual.x), escalaActual.y, escalaActual.z);
150     }
151 }
```

```
1 using UnityEngine;
2 using UnityEngine.SceneManagement;
3
4 public class JefeBase : MonoBehaviour
5 {
6     public Transform jugador; // Referencia al jugador
7     public Transform[] puntosMovimiento; // Puntos de patrulla
8     public float velocidadMovimiento = 3f;
9     public float tiempoQuietoMin = 2f; // Tiempo mínimo en estado quieto
10    public float tiempoQuietoMax = 5f; // Tiempo máximo en estado quieto
11    public GameObject rayoPrefab; // Prefab del primer rayo
12    public GameObject ataqueEspecialPrefab; // Prefab del tercer ataque
13    public Transform puntoDisparo; // Punto desde donde se disparan los ataques
14    public Transform[] puntosDisparoDoble; // Spawns para el segundo ataque
15    public float tiempoEntreAtaques = 3f; // Tiempo entre cada ataque
16
17    [Range(0f, 1f)] public float probabilidadAtaque = 0.5f; // Probabilidad de que el jefe realice un ataque
18    public CanvasGroup pantallaOscura; // CanvasGroup para la transición de pantalla
19    public float duracionTransicion = 2f; // Duración de la transición de visible a oscuro
20    public string nombreSiguienteEscena; // Nombre de la escena a cargar
21
22    public int vidaMaxima = 100; // Vida máxima del jefe
23    private int vidaActual; // Vida actual del jefe
24
25    private int puntoActual = 0; // Índice del punto actual
26    private Animator animator;
27
28    private bool estaQuieto = false; // Indica si el jefe está quieto
29    private float tiempoQuieto; // Duración actual del estado quieto
30    private float tiempoAtaque; // Control de tiempo entre ataques
31    private bool puedeAtacar = false; // Controla si el jefe puede atacar
32
33    private void Start()
34    {
35        animator = GetComponent<Animator>();
36        jugador = GameObject.FindGameObjectWithTag("Player").transform; // Buscar al jugador por su tag
37        vidaActual = vidaMaxima; // Inicializa la vida del jefe
38        EstablecerQuieto(); // Iniciar en estado quieto
39    }
}
```

## Scripts de seguimiento de cámara al personaje (Cámara - Amaru)

```

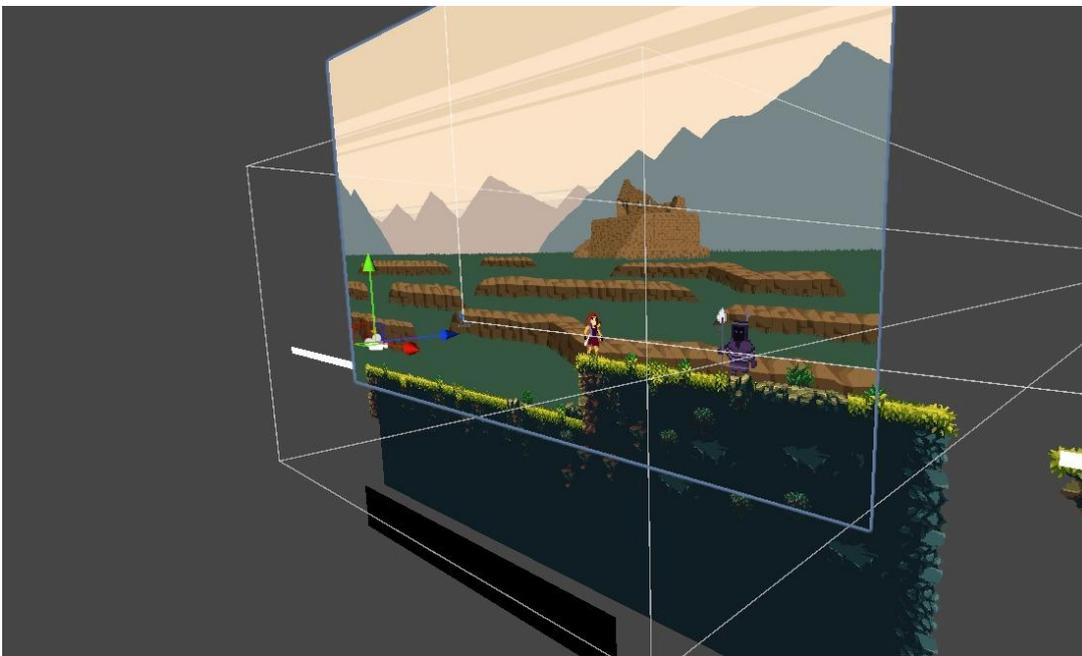
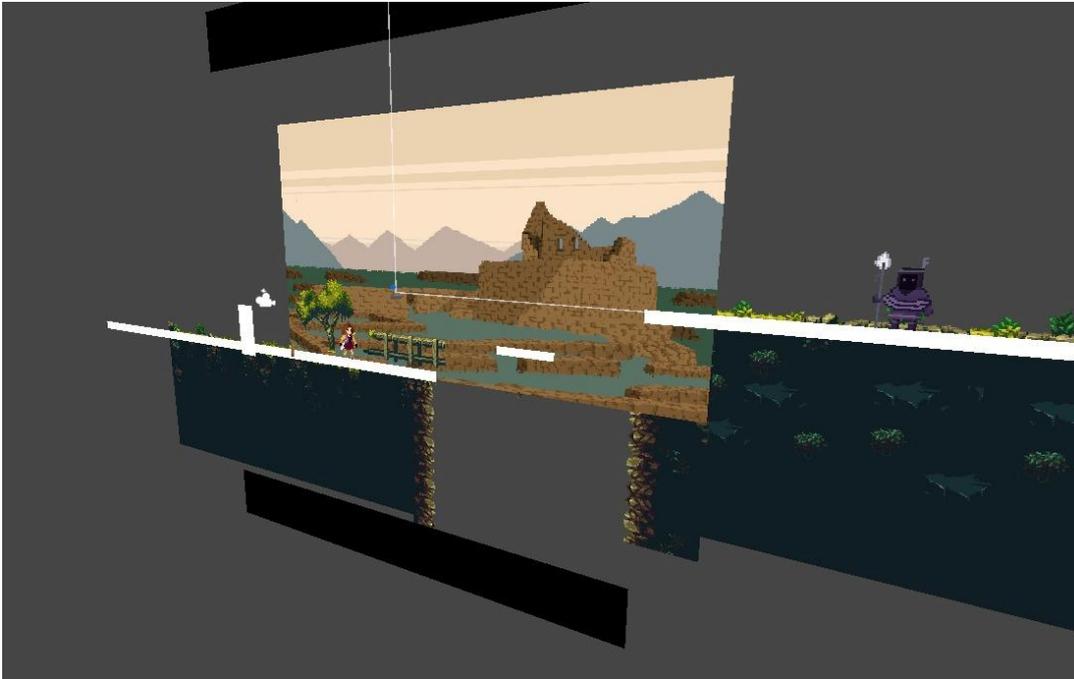
13     private Vector3 posicionSuavizada; // Posición suavizada para evitar movimientos bruscos
14     private float alturaBase; // La altura más baja que la cámara seguirá
15
16     void Start()
17     {
18         // Inicializar la altura base en la posición inicial de la cámara
19         alturaBase = transform.position.y;
20     }
21
22     void LateUpdate()
23     {
24         // Calcular la posición deseada en función del personaje y el offset
25         posicionDeseada = new Vector3(personaje.position.x + offset.x, transform.position.y, transform.position.z);
26
27         // Ajustar la posición vertical si el personaje supera la altura base + umbral
28         if (personaje.position.y > alturaBase + alturaUmbral)
29         {
30             alturaBase = personaje.position.y - alturaUmbral; // Actualizar la altura base
31         }
32         else if (personaje.position.y < alturaBase) // Si baja más de lo visible, la cámara sigue
33         {
34             alturaBase = personaje.position.y;
35         }
36
37         // Restringir la posición deseada dentro de los límites definidos
38         posicionDeseada.x = Mathf.Clamp(posicionDeseada.x, limiteInferiorIzquierdo.x, limiteSuperiorDerecho.x);
39         posicionDeseada.y = Mathf.Clamp(alturaBase, limiteInferiorIzquierdo.y, limiteSuperiorDerecho.y);
40
41         // Suavizar el movimiento de la cámara
42         posicionSuavizada = Vector3.Lerp(transform.position, posicionDeseada, suavizado);
43
44         // Aplicar la posición suavizada
45         transform.position = posicionSuavizada;
46     }
47 }
48

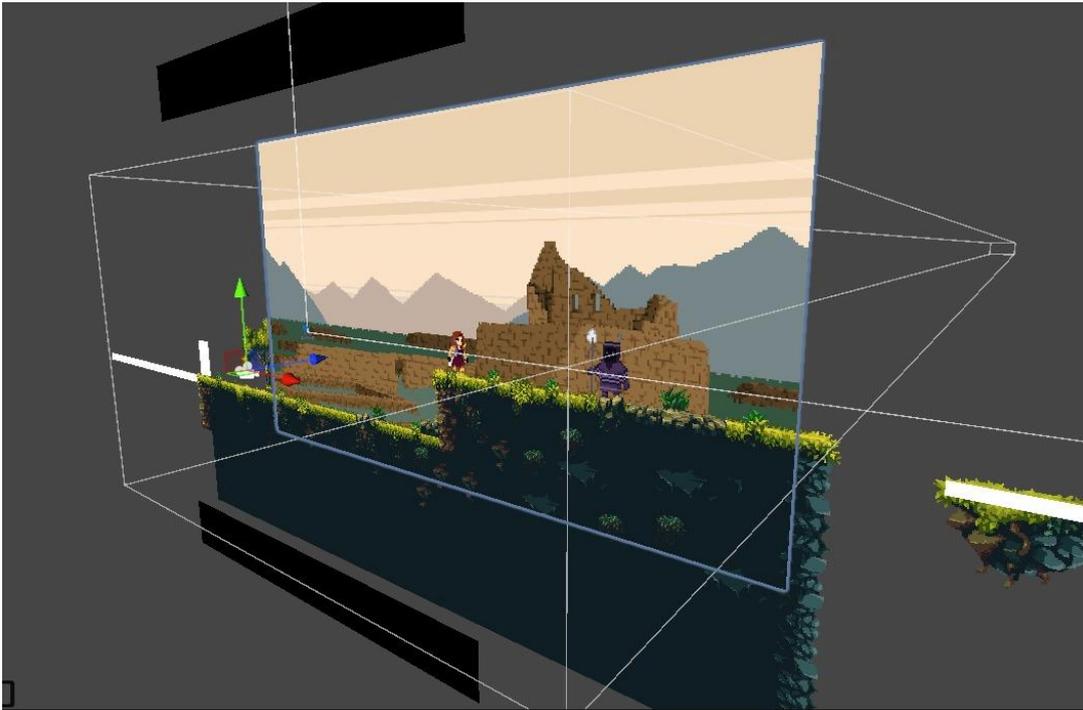
```

```

1 using UnityEngine;
2
3 public class CamaraConLmites : MonoBehaviour
4 {
5     public Transform personaje; // Referencia al personaje
6     public Vector2 offset; // Distancia entre la cámara y el personaje en X e Y
7     public Vector2 limiteInferiorIzquierdo; // Límite inferior izquierdo de la cámara
8     public Vector2 limiteSuperiorDerecho; // Límite superior derecho de la cámara
9     public float suavizado = 0.125f; // Suavidad del movimiento de la cámara
10    public float alturaUmbral = 2f; // Altura mínima que el personaje debe superar para que la cámara suba
11
12    private Vector3 posicionDeseada; // Posición deseada de la cámara
13    private Vector3 posicionSuavizada; // Posición suavizada para evitar movimientos bruscos
14    private float alturaBase; // La altura más baja que la cámara seguirá
15
16    void Start()
17    {
18        // Inicializar la altura base en la posición inicial de la cámara
19        alturaBase = transform.position.y;
20    }
21
22    void LateUpdate()
23    {
24        // Calcular la posición deseada en función del personaje y el offset
25        posicionDeseada = new Vector3(personaje.position.x + offset.x, transform.position.y, transform.position.z);
26
27        // Ajustar la posición vertical si el personaje supera la altura base + umbral
28        if (personaje.position.y > alturaBase + alturaUmbral)
29        {
30            alturaBase = personaje.position.y - alturaUmbral; // Actualizar la altura base
31        }
32        else if (personaje.position.y < alturaBase) // Si baja más de lo visible, la cámara sigue
33        {
34            alturaBase = personaje.position.y;
35        }
36
37        // Restringir la posición deseada dentro de los límites definidos
38        posicionDeseada.x = Mathf.Clamp(posicionDeseada.x, limiteInferiorIzquierdo.x, limiteSuperiorDerecho.x);
39        posicionDeseada.y = Mathf.Clamp(alturaBase, limiteInferiorIzquierdo.y, limiteSuperiorDerecho.y);

```





### Anexos 5.- (Elaboración del videojuego - funcionalidad)

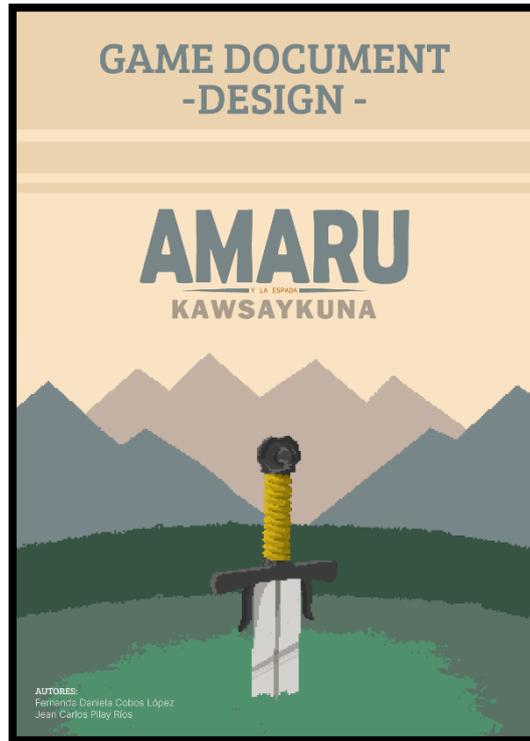


Fin del juego

Salir al Menú Principal



**Anexo 6.- (Prototipado - Diseño del Game Document Design del videojuego)**



**CONTENIDO**

<b>Concepto</b>	
Introducción	1.1
Concepto general	1.2
Resumen	1.3
Propósito	1.4
Público objetivo	1.5
Género	1.6
Plataformas	1.7
Referencias	1.8
<b>Mecánicas</b>	
Mecánicas de exploración	3.1
Mecánicas de combate	3.2
Niveles y misión principal	3.3
Controles	3.4
<b>Enemigos</b>	
Enemigo principal	5.1
Enemigos genéricos	5.2
<b>Historia</b>	
Resumen del guión	7.1
<b>Audios</b>	
Música y sonidos	9.1
<b>Arte</b>	
2.1	Aspecto general
2.2	Estilo de personaje
2.3	Aspectos de los enemigos
2.4	Entorno
2.5	Interfaz
2.6	Assets
2.7	Inspiración
<b>Protagonista</b>	
4.1	¿Quién es la protagonista?
4.2	Ficha de personaje principal
4.3	Estados y animaciones
4.4	Combate
4.5	Armas
<b>Entornos - Niveles</b>	
6.1	Mitad del Mundo
6.2	Quilotoa
6.3	Ingapirca
<b>Interfaz</b>	
8.1	Cámara
8.2	GUI
8.3	HUD
<b>Monetización</b>	
10.1	Monetización

## Concepto

### 1.1 Introducción

En el corazón de los Andes ecuatorianos, se esconde una historia que conecta la cultura y la fantasía. Amaru y la espada Kawsaykuna, es un videojuego 2D realizado con un estilo pixel art de acción y aventura que lleva a los jugadores a un viaje a través de la cultura Inca, explorando sus misterios, paisajes y deidades. Este juego, diseñado para ser una experiencia tanto educativa como entretenida, presenta un mundo en el que la riqueza cultural del Ecuador cobra vida a través de la narrativa.

El juego sigue a Amaru, una joven valiente que, al desenterrar una espada ancestral en un rincón remoto del bosque, desata una antigua maldición que libera a un poderoso Dios Inca corrompido por la oscuridad, el Dios Inti. Con la espada Kawsaykuna en mano, Amaru debe recorrer tres de los lugares más emblemáticos de la sierra ecuatoriana: las ruinas de Ingapirca, la majestuosa laguna de Quilotoa y la icónica Mitad del Mundo en Quito. Durante su viaje, los jugadores no solo enfrentarán a la deidad mencionada, sino que también tendrán la oportunidad de explorar los paisajes turísticos, acompañados de información histórica y cultural que enriquece la experiencia.

A diferencia de otros videojuegos, Amaru y la espada Kawsaykuna se enfoca exclusivamente en estos tres niveles principales, eliminando misiones secundarias para dar protagonismo total a la narrativa, los paisajes y el enfrentamiento con el Dios Inti.



### 1.2 Concepto general

Amaru y la espada Kawsaykuna, es un videojuego 2D hecho con estilo pixel art de acción y aventura que combina la exploración y el combate en un viaje por tres escenarios claves de la sierra ecuatoriana: Ingapirca, Quilotoa y Quito. Los jugadores asumirán el rol de Amaru, una joven destinada a restaurar el equilibrio en el mundo tras liberar, sin querer, una antigua maldición que ha corrompido al Dios Inti. Cada nivel ofrece una experiencia única, con paisajes deslumbrantes, carteles informativos sobre los lugares turísticos y una batalla final con la deidad mencionada. El juego está diseñado para resaltar la riqueza cultural y paisajística del Ecuador mientras ofrece una experiencia inmersiva y educativa.

## Arte

### 2.1 Aspecto general

El diseño visual de Amaru y la espada Kawsaykuna está cuidadosamente construido para destacar la riqueza cultural y paisajística de la sierra ecuatoriana, transportando al jugador a dos lugares icónicos que fueron importantes para el imperio Inca en Ecuador: las ruinas de Ingapirca y la Mitad del Mundo en Quito; la laguna de Quilotoa es parte de uno de los niveles. Sin embargo, los Incas no se asentaron en ese sitio, Quilotoa forma más parte de los niveles por ser un lugar turístico inolvidable por muchos extranjeros. Cada nivel ofrece una experiencia visual única con elementos distintivos que conectan la narrativa del juego con su entorno.



### 2.2 Estilo del personaje

El juego, diseñado en estilo pixel art, presenta personajes con un estilo simple y clásico característico de esta técnica. Amaru, la protagonista principal, está inspirada en las características de una joven ecuatoriana. Su cabello marrón oscuro y su vestido púrpura reflejan dinamismo y misterio, mientras que su espada dorada resalta su conexión con la narrativa. El diseño de Amaru combina elementos modernos y tradicionales, fortaleciendo su vínculo con la temática incaica que define el juego.



### 2.3 Aspecto de los enemigos

El videojuego cuenta con enemigos principales y un enemigo genérico. El enemigo genérico fue diseñado basado en la apariencia de una persona perteneciente al ejército incaico, siguiendo el estilo pixel art. Los creadores lo simplificaron sin perder su esencia ni alejarse de su apariencia. También, este enemigo aparecerá en cada nivel para aumentar la dificultad para la protagonista.



Además, el videojuego cuenta con un enemigo principal que, al igual que la protagonista, fue creado al estilo pixel art, pero basado en el Dios Inca Inti. Se simplificó tratando de no alejarse de su apariencia original. Aunque sea un enemigo corrompido por el mal, se optó por no usar tonos oscuros. Se usaron tonos cálidos, como amarillo y rojo, propios del dios Inti. Además, sus colores fueron pensados para que no se confundieran con los usados en los fondos del escenario.



#### 2.4 Entorno

Los entornos en Amaru y la espada Kawsaykuna están diseñados en estilo pixel art, destacando la riqueza cultural y natural de la sierra ecuatoriana. Cada nivel refleja un lugar icónico: la laguna de Quilotoa con tonos tierra y azules vibrantes que resaltan la laguna rodeada de tierra; las ruinas de Ingapirca, representadas con texturas marrones, tonos propios de una ruina, que evocan su legado incaico, enmarcadas por un fondo de montañas; y la Mitad del Mundo, con su monumento que hace contraste con el cielo azul y la vegetación verde, resaltando su relevancia cultural. La composición de cada entorno dirige la atención hacia los elementos principales, integrando detalles como vegetación y caminos para enriquecer la inmersión. Estos paisajes, además de su atractivo visual, incluyen carteles informativos que refuerzan el valor educativo del juego.

#### 2.5 Interfaz

En la elaboración de la interfaz se buscó un diseño simple, una interfaz fácil de entender. Se puede observar la barra de vida del jugador y las cantidades de oportunidades que tiene por nivel. Al principio de cada nivel se puede leer el nombre del sitio donde se encuentra.



#### 2.6 Assets

El videojuego incluye diversos assets, los cuales ayudan a que Amaru pueda continuar con su trayectoria en cada nivel. Algunos de estos assets simulan el suelo, y que estos tengan apariencia a césped con rocas debajo de ellos. Estos son inspirados en assets encontrados en otros videojuegos.

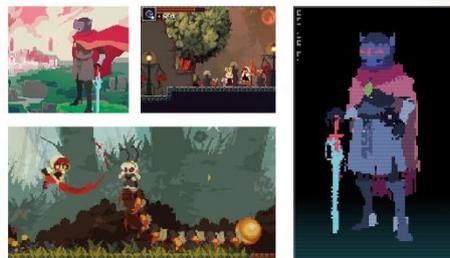


El cartel es un elemento fundamental que proporciona información al usuario. En él, se detallan los controles del juego, como caminar, saltar, entre otros; además de datos sobre el nivel en el que se encuentra el jugador. En el último nivel, además de ofrecer información sobre el lugar, el cartel incluye detalles sobre el Dios Inti, el enemigo principal al que el jugador deberá enfrentarse.



#### 2.7 Inspiración

La inspiración para la creación del diseño del videojuego fue netamente basado en el estilo pixel art y los paisajes del Ecuador. Los videojuegos Momodora y Hyper Light Drifter sirvieron como inspiración.



## Protagonista

### 4.1 ¿Quién es la protagonista?

Amaru es una joven valiente y decidida. Su diseño en pixel art está basado en características de una joven ecuatoriana. Su cabello marrón oscuro y su vestido púrpura. La espada dorada que la acompaña es más que un arma: es un símbolo de su destino y la clave para restaurar el equilibrio en un mundo sumido en el caos. Amaru representa el coraje humano frente a lo desconocido, combinando en su apariencia elementos modernos



### 4.2 Ficha de personaje principal

Nombre: Amaru  
Edad: 22 años  
Sexo: Femenino

### 4.3 Estados y animaciones

Nuestra personaje principal consta de acciones básicas, como las siguientes:

**Idle:** De pie y relajada, hace un leve movimiento hacia arriba con su torso cuando respira moviendo ligeramente sus brazos. La animación idle se puede apreciar cuando el personaje se encuentra en un punto fijo.



Animación de idle - Amaru.

**Agacharse:** Dobra ambas piernas hasta obtener una posición que le brinde equilibrio en una posición más baja.



Animación de agacharse - Amaru.

**Correr:** Mueve cada pierna alternándolas para dar la sensación de movimiento y avance o retroceso.



Animación de correr - Amaru.

**Caer:** Estira ambas piernas, una más que la otra; movimiento ligero de las vestimenta y cabello.



Animación de caer - Amaru.

**Saltar:** Flexiona y estira ambas piernas, una más que la otra; movimiento ligero de las vestimenta y cabello.



Animación de saltar - Amaru.

### 4.4 Combate

**Hurt-effect:** Ligero salto de retroceso con una pequeña inclinación del torso.



Animación de hurt-effect - Amaru.

**Ataque:** Hace uso de la espada, ataca con fuerza usando en su totalidad la espada; primero lleva la espada hacia atrás para ganar impulso y poder atacar.



Animación de ataque - Amaru.

**Muerte:** Estira ambas piernas, una más que la otra; movimiento ligero de la vestimenta y el cabello.



Animación de hurt-effect - Amaru.