



Universidad Internacional de La Rioja
Escuela Superior de Ingeniería y Tecnología

Máster Universitario en Diseño y Desarrollo de Videojuegos

Desarrollo del videojuego “Ancient Galápagos”

Trabajo fin de estudio presentado por:	CARDONA TIPAN, RENE CARRIÓN ROSILLO, CÉSAR REYES BELTRÁN, DIEGO ALEJANDRO
Tipo de trabajo:	VIDEOJUEGO 3D
URL del repositorio de código	https://github.com/AcientGalapagos/Galapagos
URL del tráiler de presentación (si procede)	https://www.youtube.com/watch?v=ZZNTYy_kSIc
URL del demo ejecutable	Link de ejecutable - ANCIENT GALAPAGOS
Fecha:	13/09/2022

Resumen

Este trabajo desarrolla la propuesta de expresar el aprendizaje del masterado, de lo cual el lector tendrá la base documental necesaria para poder desarrollar el videojuego "Ancient Galápagos".

Así, para la ejecución de este trabajo se han utilizado metodologías relacionadas al diseño de videojuegos, lo cual ha permitido tener una visión clara de la evolución del desarrollo del proyecto mediante la asignación y control de actividades por parte de los integrantes del equipo.

El videojuego recrea un escenario basado en la teoría de la evolución, contribuyendo a generar conocimiento sobre esta temática, por tal razón, "Ancient Galápagos" está desarrollado bajo el motor Unreal Engine 4, obteniendo como resultado, diferentes recursos construidos durante el desarrollo del proyecto como son modelos 3D, texturas, blueprints (BPs), animaciones y la adaptación del código fuente.

Finalmente, el videojuego como prototipo a implementar consta de tres niveles, donde el jugador debe ir destacando sobre cada obstáculo, enemigos y saber encontrar los objetos necesarios que vayan permitiendo realizar el rescate de cada personaje y con ello avanzar a los niveles siguientes para poder concluir con éxito el fin del juego.

Palabras clave: Juego en Tercera Persona, Videojuego, Galápagos, Especies, Charles Darwin.

Abstract

This work develops the proposal to express the learning of the master's degree, from which the reader will have the necessary documentary base to be able to develop the video game "Ancient Galapagos".

Thus, for the execution of this work, methodologies related to the design of video games have been used, which has allowed a clear vision of the evolution of the development of the project through the assignment and control of activities by the members of the team.

The video game recreates a scenario based on the theory of evolution, contributing to generate knowledge on this subject, for this reason, "Ancient Galapagos" is developed under the Unreal Engine 4 engine, obtaining as a result, different resources built during the development of the project. such as 3D models, textures, blueprints, animations and the adaptation of the source code.

Finally, the video game as a prototype to be implemented consists of three levels, where the player must stand out over each obstacle, enemies and know how to find the necessary objects that allow each character to be rescued and thus advance to the following levels to be able to conclude. successfully end game.

Keywords: Third Person Game, Videogame, Galapagos, Species, Charles Darwin.

Índice de contenidos

1.1.	Ficha técnica	12
1.2.	Público Objetivo	13
1.3.	Objetivos del juego	13
1.4.	Referencias	13
1.4.1.	The Legend of Zelda: The Wind Waker	13
1.4.2	Snowcraft	15
1.4.3	Tunic	15
2	El concepto del videojuego	17
2.1	Ambientación	17
2.2	Mecánicas principales	17
2.3	Dinámicas principales	18
2.4	Diseño artístico	19
2.5	Resumen de la historia	21
3	Planificación del proyecto	21
3.1	Preproducción	25
3.2	Producción	26
3.2.2	Actividades del proyecto	27
3.2.3	Cronograma de Actividades	27
3.3	Postproducción	29
3.4	Roles del proyecto	29
4	Diseño Arquitectónico del Software	31
4.1	Motor de programación	31
4.2	Diagrama de flujo de interacciones con la arquitectura	32

4.3	Motor de sonido	34
4.4	Motor de físicas	34
4.5	Motor gráfico	34
4.6	Blueprints	34
4.7	Flujo de importación de modelos desde blender, mixamo y unity	35
4.8	Estructura de carpetas del proyecto	36
4.9	Diagrama modular (capa principal)	36
4.10	Implementación	37
4.10.1	Implementación del Escenario Isla Isabela	50
4.10.2	Implementación de mecánicas básicas de INTI	57
4.10.3	Implementación del sistema de Audio	59
4.10.4	Implementación del sistema de recogida de objetos	60
4.10.5	Implementación del sistema de SPAWN	60
4.10.6	Implementación del sistema de resolución de enigmas	61
4.10.7	Implementación del sistema de Regeneración del personaje cuando muere y guardado de checkpoints.	61
4.10.8	Implementación de la IA de los enemigos	62
4.10.9	Implementación del sistema de teletransporte	63
4.10.10	Implementación del mapa dinámico	64
4.10.11	Implementación del sistema de puntuación	65
4.11	Control de versiones - GitHub	66
5	GDD de alto rendimiento	67
5.1	Introducción	67
5.2	Alcance del juego	67
5.3	Estilo visual de alto nivel	68

5.4	Interfaz	68
5.5	Flujo general	71
5.6	Diseño de niveles	73
5.6.1	Nivel 1	74
5.6.1.1	Inicio del juego, Nivel 1, pantalla principal, acción 1	74
5.6.1.2	Nivel 1, misión 1, liberar las especies	76
5.6.1.3	Transición, Nivel 1, pantalla principal, acción 2	81
5.6.1.4	Nivel 1, misión 2, rescate George	81
5.6.1.5	Nivel 1, misión 3, Carrera por alcanzar la Fuente Mágica	89
5.6.2	Nivel 2	91
5.6.3	Nivel 3	92
5.7	El jugador	93
5.7.1	Comportamiento	93
5.7.2	Combate	94
5.7.3	Armas	95
5.7.4	Vidas	95
5.8	NPCs (Nativos de la isla)	95
5.8.1	Comportamiento	96
5.8.2	Combate	97
5.8.3	Armas	97
5.8.4	Vidas	97
5.9	Enemigos – Piratas	97
5.9.1	Comportamiento	97
5.9.2	Combate	97

5.9.3	Armas	98
5.9.4	Vidas	98
5.10	Otras mecánicas	98
5.10.1	Cajas con especies para rescatar	98
5.10.1.1	Sistema de puntuación	99
5.10.1.2	Spawn de Cajas	100
5.10.1.3	Cajas móviles	100
5.10.2	Pulsadores	100
5.10.2.1	Sistema de puntuación	100
5.10.2.2	Detalle de la mecánica para pulsadores individuales	100
5.10.2.3	Detalle de la Mecánica de pulsadores conjuntos	100
5.10.3	Bloques / Obstáculos	101
5.10.3.1	Mecánica en Obstáculos/bloques básicos	101
5.10.3.2	Mecánica en Obstáculos/bloques destruibles	101
5.10.3.3	Mecánica en Obstáculos/bloques móviles	101
5.10.3.4	Mecánica en Obstáculos/puertas e interruptores	102
5.11	GamePlay y dinámicas	103
5.11.1	Mecánicas del Nivel 1 – Misión 1 - Rescate de especies	110
5.11.2	Mecánicas del Nivel 1 – Misión 2 - Escape room	112
5.11.3	Mecánicas del Nivel 1 – Misión 3 - Esquiva Obstáculo	114
6	Conclusiones	116
7	Postmorten del juego	117
7.1	Falta de experiencia en programación Unreal	117
7.2	Falta de personal que integre el equipo	117

7.3 Experiencia de usuario	117
BIBLIOGRAFIA	119

Índice de figuras

Figura 1. The legend of Zelda: The Wind Waker.....	14
Figura 2. Snowcraft.....	15
Figura 3. TUNIC.....	16
Figura 4. Propuesta visual de diseño artístico.	19
Figura 5. Moodboard niveles.	19
Figura 6. Moodboard personaje principal.....	20
Figura 7. Moodboard NPCs.	20
Figura 8. Moodboard NPCs Enemigos.....	20
Figura 9. Props & Assets.....	21
Figura 10. Roles.	22
Figura 11. Activos.	23
Figura 12. Licencias.	23
Figura 13. Man Month Cost.	23
Figura 14. Fase preproducción.....	24
Figura 15. Fase producción.	24
Figura 16. Fase postproducción.	25
Figura 17. Diagrama Gantt. (Flujo).....	29
Figura 18. Diagrama de arquitectura de software.....	33
Figura 19. Flujo importación de modelos.	35
Figura 20. Esqueleto FBX.....	35

Figura 21. Estructura de carpetas de los Recursos empleados.....	36
Figura 22. Framework Class Relationships.....	37
Figura 23. Esquema general de implementación.....	38
Figura 24. Esquema implementación MODO DE JUEGO.	39
Figura 25. Esquema general de recursos y animaciones desarrollados e importados.....	40
Figura 26. Esquema general de implementación de la interfaz de usuario.....	42
Figura 27. Esquema general de implementación de la zona ISABELLA.	43
Figura 28. Esquema general de implementación de la MISION 1 – RESCATE DE ESPECIES. ...	44
Figura 29. Esquema general de implementación de la MISION 2 – RESCATE DE GEORGE, fases 0 y 1.	45
Figura 30. Esquema general de implementación de la MISION 1 – RESCATE DE GEORGE fases 2, 3 y 4.	46
Figura 31. Esquema general de implementación de la MISION 3.....	47
Figura 32. Variable NIVELMISIONFASE y su codificación.	49
Figura 33. Configuración del terreno para la Isla Isabela.....	51
Figura 34. Multiplicación de las coordenadas por 0 y asignadas a las UVS.	52
Figura 35. Creación del material dinámico del mar.	54
Figura 36. Aspecto Lowpoly del mar.	55
Figura 37. Movimiento del agua del mar.	56
Figura 38. BP animaciones (Anim Graph).....	58
Figura 39. Máquina de estados de la animación Inti.	59
Figura 40. Componente de sonido.	59
Figura 41. Implementación del mapa dinámico.....	64
Figura 42. HUD Debug para el videojuego.	70
Figura 43. Flujo del Nivel 1 – Misión 1.	71

Figura 44. Flujo del Nivel 1 – Misión 2 y 3.....	72
Figura 45. Flujo de Nivel 2 y 3 (No implementado).....	73
Figura 46. Mapa de la Isla Isabella.	75
Figura 47. Mapa de la misión 1 - fase 1 del Nivel 1.....	77
Figura 48. Mapa de la Misión 1 – fase 2 del Nivel 1.....	79
Figura 49. Mapa de la Misión 1 – fase 3 del Nivel 1.....	80
Figura 50. Mapa de la Misión 2 – fase 0 del Nivel 1.....	83
Figura 51. Mapa de la Misión 2 – fase 1 del Nivel 1.....	84
Figura 52. Mapa de la Misión 2 – fase 2 del Nivel 1.....	86
Figura 53. Mapa de la Misión 2 – fase 3 del Nivel 1.....	87
Figura 54. Mapa de la Misión 2 – fase 4 del Nivel 1.....	89
Figura 55. Resumen de la Misión 3.	91
Figura 56. Nivel 1, misión 1 rescate.	103
Figura 57. Zonas de rescate de especies.....	104
Figura 58. Zona de rescate de lobos marinos.	105
Figura 59. Zona de rescate iguanas.....	106
Figura 60. Zona rescate de tortuga.	106
Figura 61. Nivel 1, Acción 2 Inti.....	107
Figura 62. Inicio y fin de oleadas.....	107
Figura 63. Nivel 1, misión 2 rescate.	108
Figura 64. Zona del pantano.....	109
Figura 65. Resolver puzzle.....	110

Índice de tablas

Tabla 1. Ficha técnica	12
Tabla 2. Mecánicas principales de los personajes.	17
Tabla 3. Cronograma de Actividades.	27
Tabla 4. Roles del proyecto.	30
Tabla 5. Descripción de las variables en el BP_Inti.	49
Tabla 6. Descripción de las variables en el BP_GM_terceraPersona.	50
Tabla 7. Descripción de las variables en el BP_Widget.	50
Tabla 8. Inputs del jugador.	57
Tabla 9. Variables BP_Enemigo.	62
Tabla 10. Variables del Bp_oleadasenemigos.	63
Tabla 11. Control del videojuego.	69

1. Introducción

El siglo XXI está pasando por una etapa de continua evolución tecnológica y con ello la industria del videojuego ha ido creciendo e innovando en características y herramientas que facilitan también su distribución digital a bajos recursos para su adquisición.

Por tal razón, este trabajo plantea el desarrollo de un prototipo de videojuego del cual se considera la cultura e historia de unas de las regiones del Ecuador, siendo un atractivo al público en general al desplazar diferentes escenarios amigables para el jugador.

1.1.FICHA TÉCNICA

Por medio de este videojuego se espera llegar a desarrollar un primer prototipo de juego que incorpore mecánicas base y suficiente concreción para que a futuro se permita su escalabilidad y desarrollo completo. En la Tabla 1, se detalla la ficha técnica:

Tabla 1. Ficha técnica

Descripción	Detalle
Género	Plataforma, Aventura.
Jugadores	Uno.
Potenciales plataformas	Computadores.
Perspectiva	Desplazamiento en tercera persona
Tipo de gráficos	3D
Motor de desarrollo	Unreal Engine 4.72.2
Modelado 3D y texturizado	Blender y 3DS Max
Jugabilidad	Explorar, andar, correr, saltar, esquivar rocas y objetos que caen, e interactuar con objetos del juego para salvar a las especies endémicas de las islas, resolver acertijos, runner.

1.2.PÚBLICO OBJETIVO

La orientación del videojuego está derivada a un público de 12 años a 40 años que le apasione la exploración de escenarios como islas con enfoque caricaturesco, que quieren a más de divertirse también quieran aprender algo nuevo.

Además, el videojuego es adecuado para un perfil de jugador tipo Midcore, e incluye la posibilidad que se pueda usar en entornos educativos a través conceptos GBL (Game Base Learning), teniendo una calificación de edades: ESRB E10+ ó PEGI 12.

1.3.OBJETIVOS DEL JUEGO

El objetivo del juego es sumergir al jugador en un mundo 3D lowpoly a escala de las islas Galápagos, mientras se genera conciencia del peligro que corren las especies, a través de un sistema de logros que agreguen una ficha por especie que el jugador haya ayudado. Una guía sobre las especies de la isla que permitirá al jugador conocer datos como su origen y hábitat.

Por medio de este videojuego se espera transmitir al jugador, la observación de paisajes, el defender las especies y crear conciencia sobre el hábitat y la evolución de la naturaleza, con la intención de que las emociones que tenga el usuario relacionen la responsabilidad de ser un héroe dentro el fabuloso entorno de las Mágicas Islas Galápagos.

1.4.REFERENCIAS

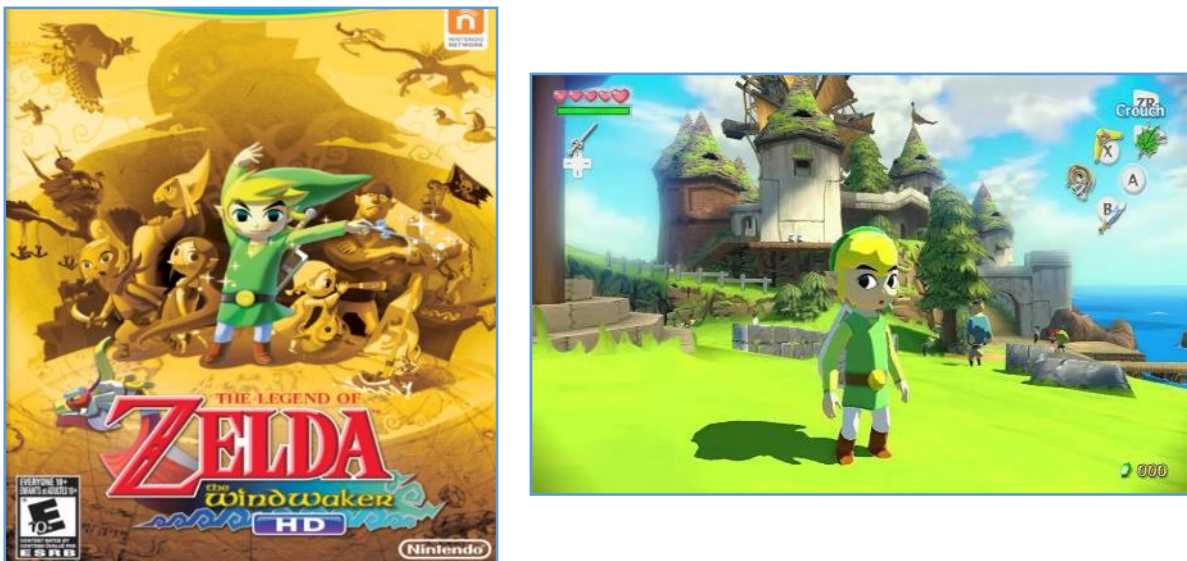
Como parte del Benchmarking inicial de este trabajo, se analizaron gran cantidad de videojuegos existentes en el mercado, de los cuales han servido como inspiración por el éxito alcanzado, por tal, hemos referenciado los videojuegos más relevantes que por sus características principales y la forma de resolución de mecánicas aportan valor a este proyecto, a continuación, se detallan los videojuegos:

1.4.1. The Legend of Zelda: The Wind Waker

Este videojuego es la décima secuela de la saga de Zelda, este juego fue desarrollado por un grupo llamado "El gran mar" siendo una de las novedades de esta entrega de la saga, su jugabilidad tiene acciones básicas para caminar, correr, atacar, defenderse y poder saltar de forma automática, también sigue utilizando el sistema de control que se viene establecido en

el juego "Ocarina of Time", el cual permite bloquear y enfocar a un enemigo o a un objetivo, se adiciona la capacidad de poder parar y cuando está enfocando a un enemigo/objetivo no ataca de manera activa ya que ciertos ataques hacen activar una señal visual, una vibración y un timbre del controlador, ese punto hace que se pueda esquivar y que realice un contraataque considerándose una táctica crucial para derrotar a los enemigos/objetivos blindados o jefes (The Legend of Zelda Wiki, 2002). En la Figura 1, se presenta la portada del juego y un ejemplo de su gameplay:

Figura 1. The legend of Zelda: The Wind Waker.



Fuente: Internet.

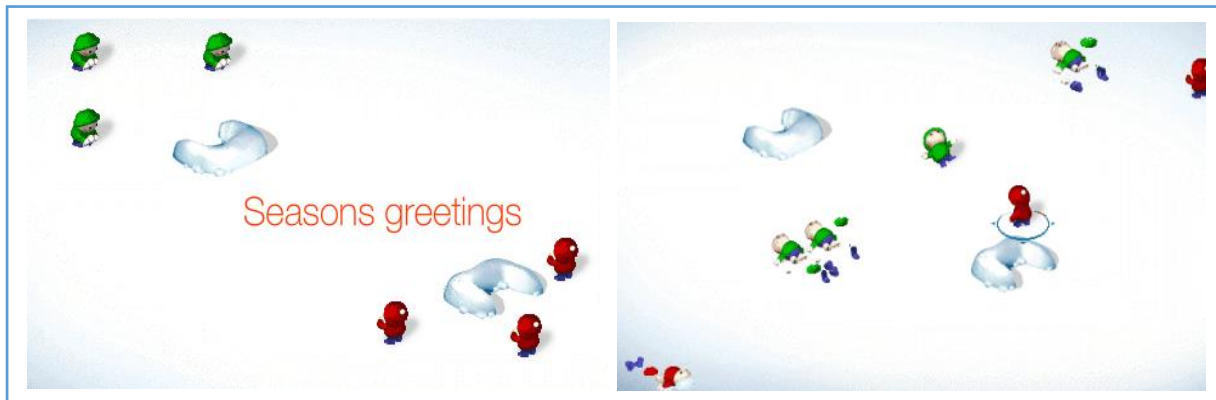
La mecánica de este juego es la principal referencia de aportación a este trabajo en cuanto el escenario de mundo abierto, la forma de transportarse de una isla a otra y el arte Low Poly heredades de sus predecesores. Por consiguiente, aprovecharemos:

- ❖ Mecánicas similares tales como: caminar en todas direcciones, saltar, interactuar con objetos, abrir puertas, ataques a distancia y melé.
- ❖ Niveles con forma de islas, proponemos crear los niveles en forma de islas, permitiendo delimitar el área de juego por el mar.
- ❖ Reemplazar el barco por una tortuga que transporte al personaje y que localice el movimiento en avanzar y detenerse.
- ❖ Localizar el uso del transporte cuando las misiones principales de la isla estén resueltas, así, se podrá embarcar en el navío.

1.4.2 Snowcraft

Este videojuego SnowCraft, se basa en el éxito que ha tenido Blizzard, siendo un juego de acción donde presenta gráficos de manera fluida, simple pero muy adictivo. Su jugabilidad es simple, se tiene que ganar una pelea de bolas de nieve de forma que vaya maniobrando a 3 niños pequeños, también, se puede mover pequeñas tropas y ajustar la potencia con lo que se dirige la bola de nieve con un click. La primera vez se enfrenta contra 3 oponentes, y a medida que vas ganando más batallas, te iras enfrentando a más oponentes cada vez más hábiles que evitarán tus bolas de nieve. (MobyGames, 2001)

Figura 2. Snowcraft.



Fuente: Internet.

Su sistema de lanzar bolitas como forma de ataque es interesante, produce una experiencia corta de satisfacción y diversión. Por consiguiente, aprovecharemos:

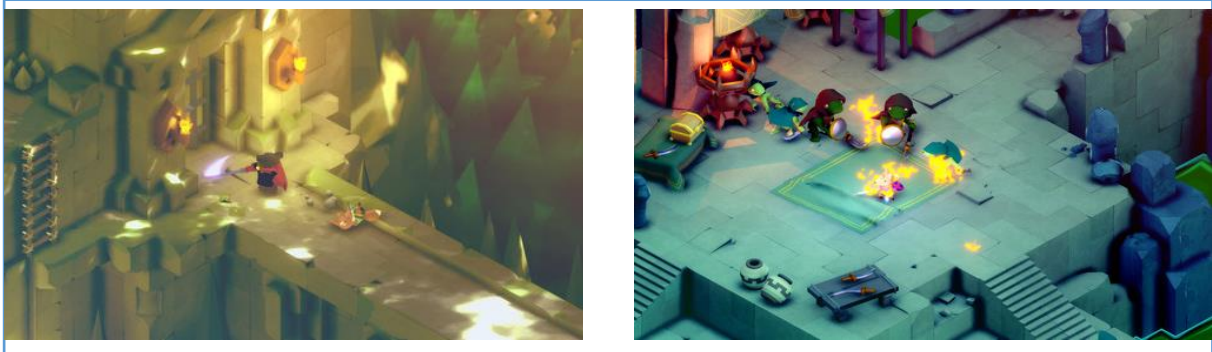
- ❖ Gráficos sencillos y en baja resolución.
- ❖ Fluidez del juego, hacer retos que sean fáciles de entender sin usar texto o tutoriales para entender la lógica del juego.
- ❖ Mecánica adictiva y retadora a medida que se avanza de niveles.

1.4.3 Tunic

Este videojuego tiende a explorar un reino que está plagado de leyendas que en el transcurso de los años han sido olvidadas, donde además se encontrarán seres con poderes ancestrales y feroces monstruos, su personaje principal del juego de acción es un zorro pequeño que emprende una aventura, siendo abandonado en una misteriosa playa por lo cual se debe

armar de valor y presto con su curiosidad, debe enfrentarse a bestias colosales, para poder ir reuniendo extraños objetos y que los mismos le desvelen secretos perdidos en el tiempo (ISOMETRICORP, 2022).

Figura 3. TUNIC.



Fuente: Internet.

Este videojuego tiene un arte de píxeles moderno y vibrante agregado de un combate ágil y rápido, también tiene una iluminación suave y cambiante con una geometría simple. Por consiguiente, aprovecharemos:

- ❖ Sus mazmorras como fuente de inspiración.
- ❖ Intentaremos trabajar los niveles para conseguir una optimización lowpoly similar.

2 EL CONCEPTO DEL VIDEOJUEGO

2.1 AMBIENTACIÓN

La ambientación de este trabajo se basa en las islas Galápagos ubicadas a 972 km de las costas del Ecuador en el océano Pacífico. Todo inicia ubicando a nuestro personaje en la primera isla llamada Isabela, años antes de su descubrimiento aproximadamente hace 500 años.

Así, los escenarios son de aspecto "Lowpoly Cartoon" que han servido para ambientar las 13 islas del archipiélago galápagos, las "chozas" o viviendas y particularidades del pueblo indígena (shuar), aquí el personaje un miembro indígena rodeado por mar y con especies mitológicas encontradas en las islas, se enfrentará con piratas afín de salvaguardarla a todo su pueblo, luchando contra enemigos que tratarán de saquear las especies de las islas, para ello, también podrá ir obteniendo habilidades mágicas que le ayudarán ir superando cada misión.

2.2 MECÁNICAS PRINCIPALES

Tenemos los tipos de mecánicas que se utilizarán tanto para el personaje principal como para los enemigos u objetivos en este videojuego, detallado al siguiente:

Tabla 2. Mecánicas principales de los personajes.

Descripción	Detalle
Las mecánicas básicas del personaje principal INTI	Parado o posición de observación, Caminar, Correr, Agacharse - Caminar lento, Saltar sobre tierra, Atacar lanzando piedras, Atacar Usando el arma arrojadiza de piedras, Esquivar ataques, Coger Objetos, Empujar Objetos, Interacción de Objetos con partes o zonas de la isla, Interacción con los NPCs, Rescate de especies, Resolución de enigmas.
Las mecánicas básicas de los NPCs	Patrullaje, Hablar, interacción con personaje principal.

Las mecánicas básicas de los
Enemigos serán:

Vigilancia: Los enemigos se detienen por un tiempo en caso de no interceptar al jugador.

Patrullaje: En caso de no detectar al jugador recorrerá los puntos de patrullaje.

Radar-Player: En ciertas partes del juego, los enemigos conocen la posición del jugador en todo momento independientemente de si es visible o no, y se desplazan a su posición.

Correr: En caso de detectar al jugador se desplaza más rápido hasta alcanzarlo.

Atacar: Al conseguir alcanzar el jugador se realiza la animación de ataque.

Fuente: Elaboración propia.

2.3 DINÁMICAS PRINCIPALES

A continuación, se describen con algo más de detalle las dinámicas del personaje principal en base a las mecánicas comentadas anteriormente:

- ❖ Sigilo, recurrente para evitar ser visto por la IA de los enemigos.
- ❖ Recorrer la isla para ello usando las habilidades de caminar, correr y saltar.
- ❖ Liberar especies llevando las cajas correspondientes a cada especie a su zona de seguridad.
- ❖ Activar acceso a zonas de la isla a través de la interacción de objetos recogidos previamente.
- ❖ Esquivar objetos como avalanchas de piedras que no nos permiten avanzar en la isla.
- ❖ Defensa de ataque.
- ❖ Ataque con piedras normales, limitado en munición.
- ❖ Ataque con piedras mágicas, ilimitada en munición.
- ❖ Ataque con palo o vara. Es el ataque por defecto, no tiene límite de ataque.
- ❖ Ataque con honda o artefacto lanza piedras, limitado en munición

2.4 DISEÑO ARTÍSTICO

El diseño artístico de este trabajo de desarrollo está enfocado en el tipo "Lowpoly Cartoon" siendo este de baja poligonización y con colores muy vivos, haciendo que los escenarios a utilizar sean más amigables, como se muestra en la Figura 4:

Figura 4. Propuesta visual de diseño artístico.



Fuente: Internet.

Por consiguiente, se detalla los moodboards desarrollados para cada nivel y para los personajes que se involucran en este videojuego:

- ❖ Moodboard por niveles referencia Isla Isabela en la Figura 5:

Figura 5. Moodboard niveles.



Fuente: Internet.

- ❖ Moodboard personaje principal y nativos referencia héroes en la Figura 6:

Figura 6. Moodboard personaje principal.



Fuente: Internet.

- ❖ Moodboard NPCs Tortugas Galápagos en la Figura 7:

Figura 7. Moodboard NPCs.



Fuente: Internet.

- ❖ Moodboard NPCs Enemigos Piratas en la Figura 8:

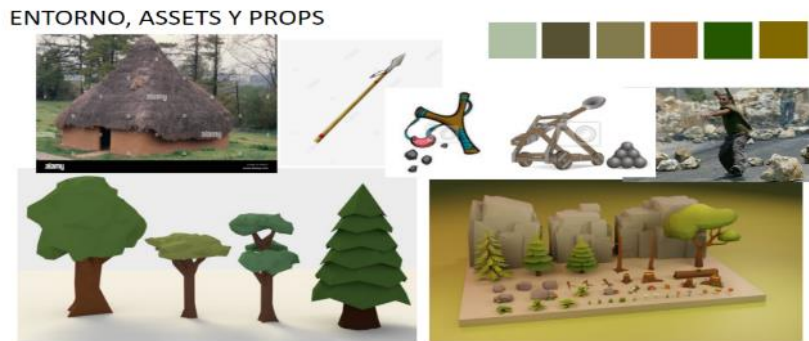
Figura 8. Moodboard NPCs Enemigos.



Fuente: Internet.

- ❖ Props & Assets en la Figura 9:

Figura 9. Props & Assets.



Fuente: Internet

2.5 RESUMEN DE LA HISTORIA

Año 1500, Las islas galápagos conocidas como las islas encantadas, el origen de su calificativo se basa en una historia que se contaba alrededor de las fogatas y pasó de generación en generación entre los nativos de la isla.

Por tal razón, la historia establece que cada vez que las especies de la isla enfrentan un peligro, los espíritus como guardianes de las especies eligen un defensor al cual probará su valentía y se le otorgan habilidades fantásticas para que pueda defender a las islas de los males que acechan a sus especies. Así, tenemos a Inti quien es un niño de aproximadamente 13 años de edad, tal vez un aldeano más de la isla isabela, pero su vida está a punto de cambiar cuando los espíritus lo elijan para ser el defensor de las especies.

Inti sumido en un mar de aventuras, derrota al pirata barba verde y encuentra entre las especies cautivas a un prisionero llamado "Sr Darwin" quien decide quedarse en las islas. Luego, Inti partió en aventura a las demás islas mientras Sr Darwin se quedó a vivir en ella ya que las especies le parecían muy interesantes y las islas, un gran lugar para vivir, Inti pensó mientras se alejaba de su isla natal en el lomo de la tortuga, tal vez el Sr Darwin podría enseñarle cosas a su aldea.

3 PLANIFICACIÓN DEL PROYECTO

Las fases del proceso de desarrollo de un videojuego varían en tamaño dependiendo de la complejidad de cada proyecto, por tal, dentro del esquema general nos encontramos con las

secciones de preproducción, producción y postproducción de donde se destaca la funcionalidad de cada fase para tener un exitoso proyecto de gestión (Maxwell, 2010).

Así, la siguiente planificación del proyecto a ejecutar como un desarrollo completo del juego, es de un prototipo de 10 minutos jugable considerados por motivos del trabajo fin de estudios, considerando a un periodo de ejecución de 15 meses integrada por una plantilla de 11 colaboradores.

❖ Recursos humanos considerados

El desarrollo del proyecto del videojuego se considera que sus fases necesitarán el siguiente tiempo:

- 3 meses de Preproducción
- 10 meses de Producción
- 1 meses de Postproducción
- 1 mes de Archivo

❖ Roles

Considerando que, en situaciones normales, diferentes recursos pueden asumir diferentes roles, es posible considerar la posibilidad de reducir la cantidad de Staff. Los roles establecidos en la Figura 10:

Figura 10. Roles.

Recurso	Coste Anual	Cantidad /meses	Total	Observaciones
STAFF				
Sr Game Designer	25.000,00 €	1	25.000,00 €	Mecanicas y hace las veces de Lead
Jr Game Designer	21.000,00 €	1	21.000,00 €	Diseño de niveles
Sr Artist	25.000,00 €	1	25.000,00 €	Se encargan también de la animación y las veces de productor
Jr Artist	21.000,00 €	1	21.000,00 €	Se encargan también de la animación
Technical Artist	25.000,00 €	1	25.000,00 €	FX Artist también
Sr Programmer	25.000,00 €	1	25.000,00 €	Hace las veces de Lead Programmer
Jr Programmer	21.000,00 €	2	42.000,00 €	
Producer	25.000,00 €	1	25.000,00 €	
QA	21.000,00 €	1	21.000,00 €	
Audio	21.000,00 €	1	21.000,00 €	Audio Fx y música

Fuente: Elaboración propia

❖ Activos

Aquí se considera al teletrabajo como principal elemento de forma que no se consideraran costes asociados al alquiler de la localidad como se muestra en la Figura 11:

Figura 11. Activos.

ACTIVOS	Los activos tecnológicos se amortizan en 4 años			
Oficina	0,00 €	0	0,00 €	Trabajo en remoto no necesario la oficina
Ordenador Diseño	1.500,00 €	2	750,00 €	
Ordenador Arte	1.500,00 €	3	1.125,00 €	
Ordenador Programación	2.000,00 €	3	1.500,00 €	
Ordenador Audio	2.000,00 €	1	500,00 €	
Ordenador Oficina	850,00 €	1	212,50 €	Comparte QA y Productor el ordenador
Pantalla	120,00 €	18	540,00 €	doble pantalla para equipos de diseño, arte, código
NAS	300,00 €	1	75,00 €	

Fuente: Elaboración propia.

❖ **Licencias** consideradas para realizar el proyecto detalladas en la Figura 12:

Figura 12. Licencias.

LICENCIAS				
Unreal	0,00 €	12	0,00 €	Unreal al ganar menos de 1 millón de dólares, lo considera gratis
Github Pro	12,00 €	15	180,00 €	15 e al mes por 15 meses de uso
Windows 11	145,00 €	12	1.740,00 €	licencias windows 11 home para todos los equipos
Office 365 (word, excel, outlook, teams, onedrive, cuentas de correo corporativa...)	120,00 €	15	1.800,00 €	licencias colaborativas Office 365, coste mensual 10 € por 12 trabajadores
Adobe Cloud (Illustrator, photoshop, substance) - licencias compartidas	750,00 €	3	2.250,00 €	diseño 2D UI y texturizado
Blender	109,00 €	2	218,00 €	Modelado 3D y animación - Herramienta similar a 3D Max, gran parte gratuita
Cubase	489,00 €	1	489,00 €	Herramienta para audio, hasta 2 millones de dolares ese valor, este valor puede ser utilizado para comprar librerías de audio y plugins

Fuente: Elaboración propia.

El total de gastos considerados nos permiten hacer un primer cálculo de "Man Month cost" representados en la Figura 13:

Figura 13. Man Month Cost.

TOTAL NETO			262.379,50 €
margen	25 %		65.594,88 €
TOTAL GROSS			327.974,38 €
TOTAL GROSS MENSUAL			27.331,20 €
MAN MONTH RATE		11	2.484,65 €

Fuente: Elaboración propia.

❖ **Fase concept y preproducción (20 mm)** en la Figura 14:

Figura 14. Fase preproducción.

		Preproduction		
		Concept	Preproduction	Preproduction
Mes		1	2	3
	<u>Artist</u>	1,0	1,0	1,0
	<u>FX Artist</u>	0,0	1,0	1,0
	<u>Animator</u>	0,0	1,0	1,0
	<u>Designer</u>	1,0	2,0	2,0
	<u>Programmer</u>	1,0	2,0	2,0
	<u>Audio</u>	0,0	1,0	0,0
	<u>QA</u>	0,0	0,0	0,0
	<u>Producer</u>	0,0	1,0	1,0
	Total			
	148	3	9	8

Fuente: Elaboración propia.

❖ Fase producción (94mm) representado en la Figura 15:

Figura 15. Fase producción.

		Production				
		Production	Production	Production	Production	Production
Mes		3	4	5	6	7
	<u>Artist</u>	1,0	1,0	1,0	1,0	1,0
	<u>FX Artist</u>	1,0	1,0	1,0	1,0	1,0
	<u>Animator</u>	2,0	2,0	2,0	2,0	2,0
	<u>Designer</u>	1,0	1,0	1,0	1,0	1,0
	<u>Programmer</u>	3,0	3,0	3,0	3,0	3,0
	<u>Audio</u>	1,0	1,0	1,0	1,0	1,0
	<u>QA</u>	0,0	0,0	0,0	0,0	0,0
	<u>Producer</u>	1,0	1,0	1,0	1,0	1,0
	Total					
	148	10	10	10	10	10

		Production			
		Production	Production	Production	Production
Mes		8	9	10	11
	<u>Artist</u>	1,0	1,0	1,0	1,0
	<u>FX Artist</u>	1,0	2,0	2,0	2,0
	<u>Animator</u>	2,0	1,0	1,0	1,0
	<u>Designer</u>	1,0	1,0	1,0	1,0
	<u>Programmer</u>	3,0	3,0	3,0	3,0
	<u>Audio</u>	1,0	1,0	1,0	1,0
	<u>QA</u>	1,0	1,0	1,0	1,0
	<u>Producer</u>	1,0	1,0	1,0	1,0
	Total				
	148	11	11	11	11

Fuente: Elaboración propia.

❖ Fase de postproducción (testing y mantenimiento) (24 mm) en la Figura 16:

Figura 16. Fase postproducción.

	Testing	Testing	Testing	Archive
Mes	12	13	14	15
Artist	1,0	1,0	1,0	1,0
FX Artist	1,0	1,0	1,0	0,0
Animator	1,0	1,0	1,0	0,0
Designer	1,0	1,0	1,0	1,0
Programmer	2,0	2,0	2,0	1,0
Audio	1,0	1,0	1,0	0,0
QA	2,0	2,0	2,0	0,0
Producer	1,0	1,0	1,0	1,0
Total				
	148	10	10	10
				4

Fuente: Elaboración propia.

En consecuencia, se transmite los roles en el desarrollo, las fases del TFE y los métodos de gestión utilizados para el adecuado seguimiento del proyecto, observación de su evolución, asignación de tareas y control de entregas para conseguir con éxito el alcanzar los objetivos planteados.

La integración de metodologías ágiles como parte principal del control y asignación de tareas, mantiene los aspectos de realizar reuniones semanales de todo el equipo y también se mantiene breves reuniones de 5 minutos diarias para asegurar que no hay problemas en el trayecto, y en caso de encontrarse, derivar recursos para asegurar alcanzar los hitos (Keith, 2010).

3.1 PREPRODUCCIÓN

En la etapa de preproducción se aborda de manera principal el objetivo del proyecto, por tal es de vital importancia el formar integrantes del equipo, y, tras varias reuniones, se fueron perfilando los roles que cada uno podría asumir dentro del proyecto.

Con la idea inicial se definió y estableció las principales mecánicas sobre las cuales se construirá el videojuego, considerando importante la parte de arte y diseño a priori para poder conseguir contenido propio diferenciador (independientemente que se evaluará la posibilidad de incorporar recursos adquiridos o bien desde el Asset Store de Unity o Unreal).

Por consiguiente, como punto de discusión se tomó las herramientas de trabajo, diferenciadas entre técnicas y colaborativas, así también, el uso multiplataforma Blender como modelador,

Unreal Engine como motor de desarrollo, GitHub como repositorio y Trello como gestor de tareas, empleando TABLEROS y tareas asignadas específicamente a cada uno de los componentes y marcando fechas de entrega.

3.2 PRODUCCIÓN

El presente TFE, tiene un alcance limitado en la producción debido primero a una limitación temporal, una limitación de competencias y una limitación de recursos.

El TFE pretende ser un demostrador que permita visualizar una representación gráfica del texto aquí referido.

En esta fase del proyecto las bases y herramientas de trabajo en equipo consistieron en el desarrollo de los documentos de diseño que fueron necesarios para el desarrollo del videojuego. De manera inicial se crearon documentos interactivos Pitch-Doc, así como Moodboards que permitían que el equipo artístico y diseño puedan comenzar a modelar los primeros elementos generando el prototipo inicial y gameplay.

Además, con el uso de la herramienta de planificación Trello, permitió realizar una distribución de actividades y tareas a quienes conforman el equipo de desarrollo de este trabajo, así, el cronograma establecido permite llevar un control de forma que se identifica los retrasos, cuellos de botella o errores inesperados que puedan suscitar.

Del mismo modo, las metodologías ágiles han permitido garantizar que este proyecto realice entregas funcionales y jugables con mayor calidad y aplicando tiempos reducidos consiguiendo flexibilidad del proyecto. Por tal razón, las metodologías ágiles han aportado a este trabajo los siguientes valores:

- ❖ La planificación se ha realizado en base a historias de usuario el cual se ha definido rotativamente de forma semanal por el equipo de trabajo.
- ❖ Estas historias se trasladan a tareas que se reflejan en Trello por lo cual mediante la planificación establecida cada integrante puede aportar de forma significativa.
- ❖ Asignación de métricas en desarrollo del proyecto, han servido para identificar y temporizar las entregas coordinadamente.

- ❖ Con las conversaciones diarias en lapsos efectivos se ha conseguido identificar dificultades en tareas con el objetivo de decidir que tarea quedan paradas, y cuales continúan en marcha.
- ❖ La generación de Sprints permiten alcanzar objetivos concretos y entregas más eficientes y de calidad.

El control de la etapa de producción de este trabajo fue muy crítico por los diversos cambios que se produjeron, sin embargo, por medio de la metodología ágil Scrum permitió la gestión de mejora de producto por medio de los requerimientos y diseños establecidos por un equipo de trabajo más adaptado al cambio de forma habitual y recurrente de situaciones o comportamientos de recursos complejos de normalizar.

3.2.2 Actividades del proyecto

Se han desarrollado las siguientes actividades:

- ❖ Diseño de niveles.
- ❖ Diseño de la arquitectura del juego.
- ❖ Diseño, modelado 3D, texturizado y animación de entorno, personajes y props.
- ❖ Creación de las mecánicas básicas de Inti.
- ❖ Programación de la simulación del movimiento del oleaje del mar en low poly.
- ❖ Programación de los niveles del juego.
- ❖ Programación de entorno de pelea con piratas.
- ❖ Programación de Spawn temporizado de enemigos y especies.
- ❖ Programación de un sistema procedural para la creación de un mapa dinámico.
- ❖ Diseño de un HUD de Depuración.
- ❖ Programación del sistema de Regeneración de Inti.

3.2.3 Cronograma de Actividades

Las actividades se detallan en la Tabla 3. Cronograma de Actividades. Tabla 3:

Tabla 3. Cronograma de Actividades.

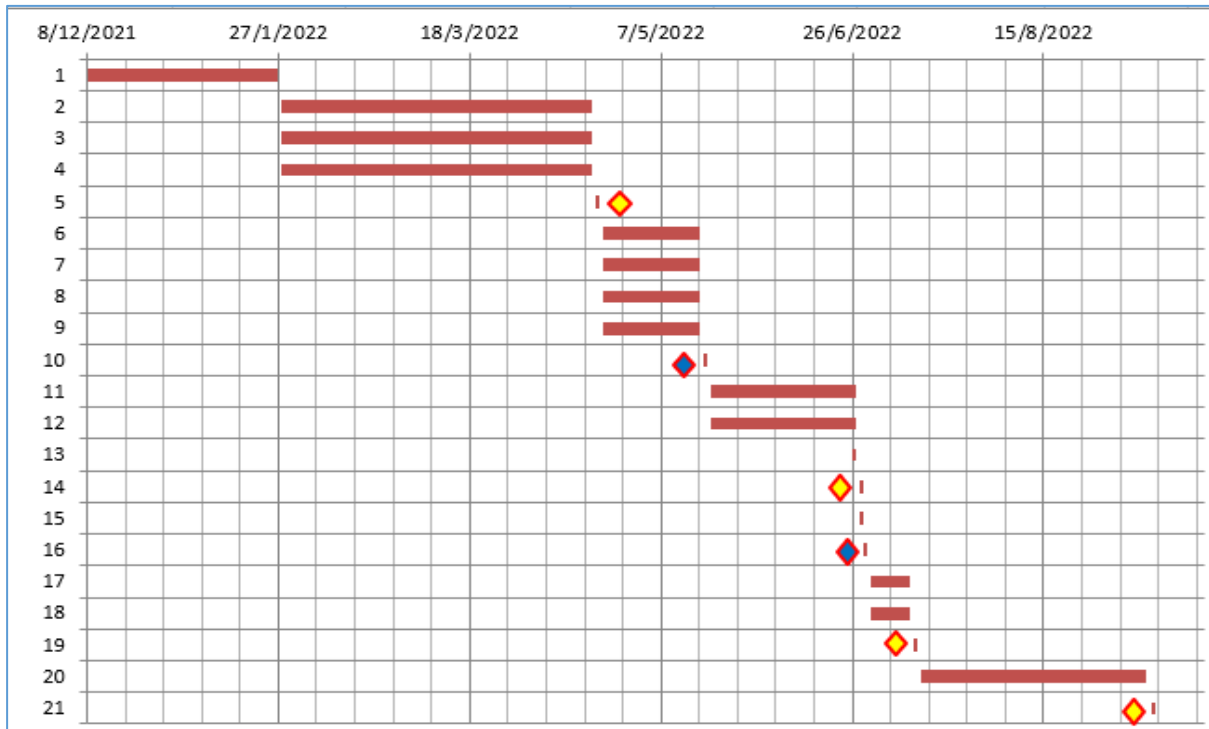
Nombre	Actividad	Fecha de inicio	Duración	Fecha de terminación	Encargado
--------	-----------	-----------------	----------	----------------------	-----------

Propuesta del TFM	1	8/12/2021	50	27/1/2022	Productores
Definir el arte del videojuego y los props	2	28/1/2022	81	19/4/2022	Artistas
Realizar el boceto de la isla en Unreal	3	28/1/2022	81	19/4/2022	Artistas
Definir las partes del documento a presentarse	4	28/1/2022	81	19/4/2022	Productores
Presentación y entrega del primer borrador	5	20/4/2022	1	21/4/2022	Productores
Diseño del gameplay inicial	6	22/4/2022	25	17/5/2022	Diseñadores
Estructuración del documento	7	22/4/2022	25	17/5/2022	Productores
Programación de los juegos Fase 1	8	22/4/2022	25	17/5/2022	Programadores
Integración de los mismos Borrador intermedio	9	22/4/2022	25	17/5/2022	Programadores
Programación de los juegos Fase 2	10	18/5/2022	1	19/5/2022	Programadores
Integración de la fase 2 de los juegos	11	20/5/2022	38	27/6/2022	Programadores
Trailer y Gameplay	12	20/5/2022	38	27/6/2022	Programadores
Reunión con el tutor	13	26/6/2022	1	27/6/2022	Diseñadores
Arreglos finales al documento	14	28/6/2022	1	29/6/2022	Productores
Borrador Final	15	28/6/2022	1	29/6/2022	Productores
Corrección de las indicaciones del tutor	16	29/6/2022	1	30/6/2022	Productores
Generar el ejecutable completo	17	1/7/2022	10	11/7/2022	Productores
Predepósito	18	1/7/2022	10	11/7/2022	Programadores
Arreglos para entrega final (memoria y gameplay)	19	12/7/2022	1	13/7/2022	Productores
Entrega Final (Extraordinaria)	20	14/7/2022	59	11/9/2022	Productores
	21	12/9/2022	1	13/9/2022	Productores

Fuente: Elaboración propia.

Presentación de flujo de actividades desde Diagrama Gantt ilustradas en la Figura 17:

Figura 17. Diagrama Gantt. (Flujo)



Fuente: Elaboración propia.

3.3 POSTPRODUCCIÓN

Este apartado del proyecto tendrá afectación como línea futura la misma que considera que este trabajo tiene los inicios de una proyección macro. Por lo tanto, la incorporación de esta etapa tendrá la base teórica guía para los casos que requiera una implementación extensa.

3.4 ROLES DEL PROYECTO

La organización de un equipo de trabajo y la asignación de roles es una etapa importante dentro de todo proyecto, ya que acorde como se va requiriendo personal para el desarrollo se debe ir dotando tanto de recurso humano como técnico donde se consolida la estructura de gestión ya sea en pymes o grandes empresas donde el equipo ya tiende a subdividirse en otros equipos dentro de un mismo proyecto (Chandler, 2020).

De este modo, este proyecto al tener poco recurso humano se podría considerar una estructura de gestión plana, donde todos tienen la misma autoridad o voz, sin embargo, no

todos tienen el mismo rol, por lo cual se detalla los roles de este equipo de trabajo en la Tabla 4:

Tabla 4. Roles del proyecto.

Descripción	Detalle
Líder Productor	Todos, Management Board
Artista Principal	Cesa Carrión, Diseño y modelado 3D
Ingeniero Líder	Rene Cardona, Programación, Codificación y estructura
Diseñador Encargado	Diego Reyes, Diseño de escenarios, flujo
Líder de Control de Calidad	Todos, calidad y gestión.

Fuente: Organigrama Adaptado de (Chandler, 2020).

4 DISEÑO ARQUITECTÓNICO DEL SOFTWARE

4.1 MOTOR DE PROGRAMACIÓN

Por el proceder del proyecto y su implementación ha llevado al equipo de trabajo a exponer sobre debate el uso de los motores multiplataforma Unity o Unreal Engine siendo dos potentes herramientas las mismas que se ha revisado sus características principales basado en su versatilidad y el lenguaje de programación.

❖ Versatilidad

Una de las primeras características consideradas a la hora de desarrollar diversos tipos de videojuegos:

2D: Los dos motores disponen de herramientas para el desarrollo de este tipo de juegos con alta calidad. No obstante, Unity dispone de un mayor número de herramientas para el desarrollo de este tipo.

3D: Los dos motores tienen unas prestaciones excelentes. En este caso, Unreal dispone de una mayor calidad del diseño gráfico.

AR: Tanto Unity como Unreal proveen esta capacidad, aunque Unity dispone de un mayor recorrido.

VR: En términos de VR Unity sobresale por encima de Unreal por sus complementos y su facilidad de integración con la infraestructura general de XR.

Multijugador: Unreal aporta soporte a este tipo de juegos. Con Unity también se puede desarrollar este tipo de juegos, aunque de momento dispone de menor experiencia que la otra opción.

Móvil: Para móvil la mejor opción es Unity.

❖ Lenguaje de programación

En cuanto a lenguaje de programación, estas herramientas utilizan en Unity el lenguaje de programación C#. Por otro lado, en Unreal Engine su lenguaje es C++, por tal razón, en relación a estos dos lenguajes de programación, C++ es considerado más complejo de aprender, sin

embargo, Unreal intenta reducir este impacto con la integración de scripters visuales llamados Blueprints.

Finalmente, después de realizar un barrido de información entre estos 2 motores expuestos la arquitectura principal para el desarrollo de este proyecto será Unreal Engine y su framework para el desarrollo en una versión 4.27.2.

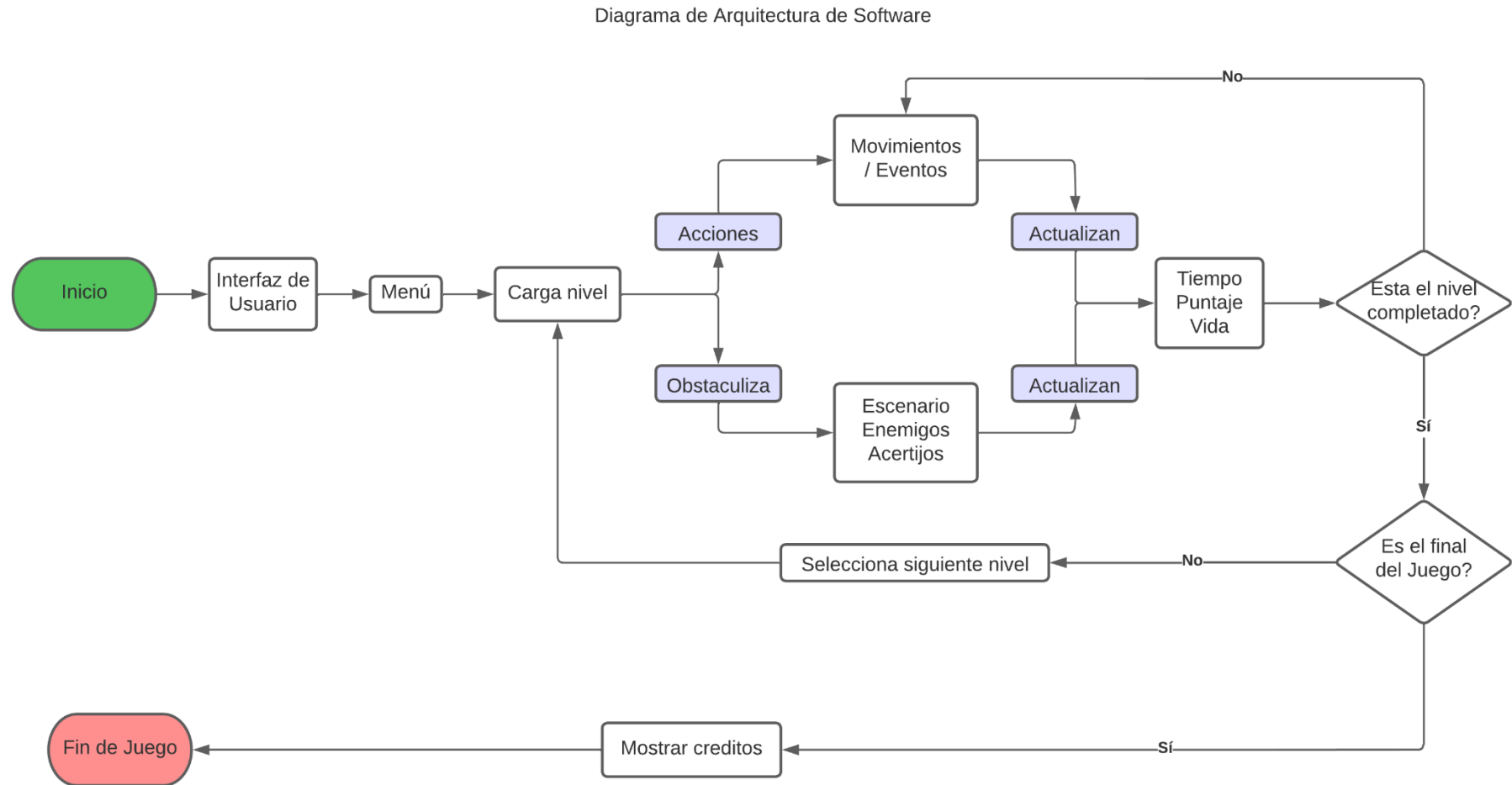
Es importante en este punto incidir en un detalle para diferenciar los diferentes elementos tal y como Unreal hace la distinción (Unreal Architecture / Unreal Documentation, n.d.) y es la diferenciación entre Actor y Objetos.

Para Unreal, Actor es toda aquella instancia que deriva de la clase Actor, mientras que los Objetos son toda aquella instancia que deriva de la clase UObject (la clase base de todos los objetos de Unreal, incluyendo Actor).

4.2 DIAGRAMA DE FLUJO DE INTERACCIONES CON LA ARQUITECTURA

El diagrama de flujo de la interacción como parte de la arquitectura, nos permitirá determinar las entradas y salidas de cada uno de los eventos, considerando al usuario como centro que inicia la aplicación, interactúa con el HUD tomando decisiones y ejecutando las mecánicas en el juego. Se presenta el diagrama de flujo del proyecto en la Figura 18:

Figura 18. Diagrama de arquitectura de software.



4.3 MOTOR DE SONIDO

Dentro de esta arquitectura es importante definir el motor del sonido, siendo importante el encargado de gestionar y reproducir cualquier audio (efectos, música, otros sonidos), Así, en el proyecto, gran parte de los sonidos son disparados por eventos y acciones del jugador, por otro lado, los demás sonidos son descargas generadas desde la tienda de Unreal Engine o creadas para cada nivel en específico, el tipo de audio que UE4 acepta es .wav a 16 bits, ningún otro tipo de extensión.

4.4 MOTOR DE FÍSICAS

El motor de físicas es imprescindible para el ajuste de las características de los escenarios a utilizar y empleado en el movimiento de cajas, en el suelo, en la resolución de enigmas, otro aspecto, está en que se generan modificaciones en la física para las siguientes observaciones como un objeto que no rote, para modificar tiempos de ejecución, para activar o desactivar la simulación de físicas y para optimizar el rendimiento en sí.

4.5 MOTOR GRÁFICO

Unreal Engine tiene un motor gráfico que no se compara con ninguno otro, siendo superior y más estable en los encargos de renderizar los recursos, así como también generando colores, sombras e iluminaciones.

Por tal razón, en este proyecto se ha trabajado con el uso de herencia de objetos para con ello reducir las llamadas al procesador y mejore el rendimiento.

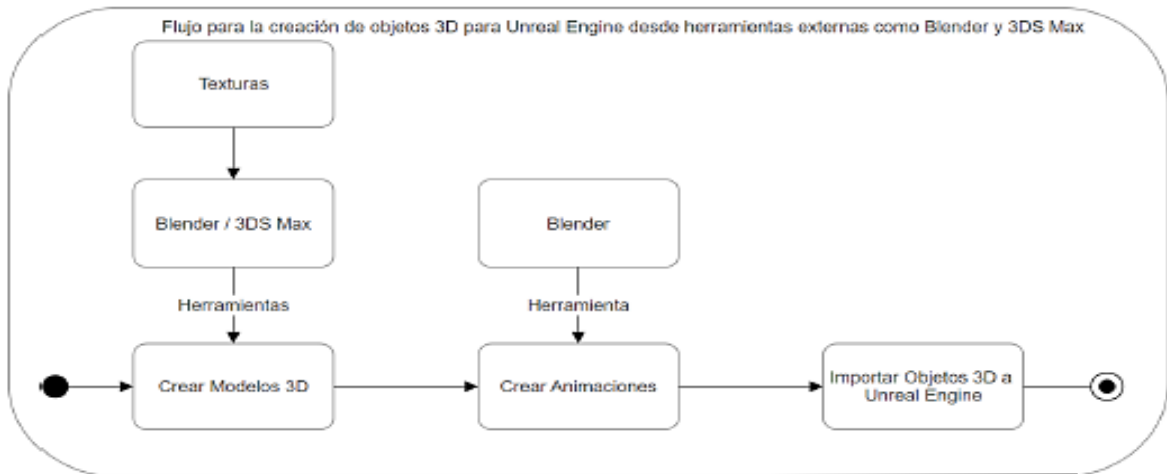
4.6 BLUEPRINTS

Con esta metodología de programación de forma visual nos permite programar las diferentes funcionalidades de forma que conectando "cajas" simulando lo desarrollado desde el diagrama de flujo de este trabajo, puesto que se vuelve imprescindible la utilización del mismo que el conocimiento de Unreal Engine ha sido el adquirido durante el proceso de este estudio académico.

4.7 FLUJO DE IMPORTACIÓN DE MODELOS DESDE BLENDER, MIXAMO Y UNITY

Durante el desarrollo se han realizado modelos propios, se han obtenido modelos del Asset Store de Unity y se han obtenido animaciones de Mixamo para los diferentes caracteres incluidos detallado un diagrama de flujo en la Figura 19:

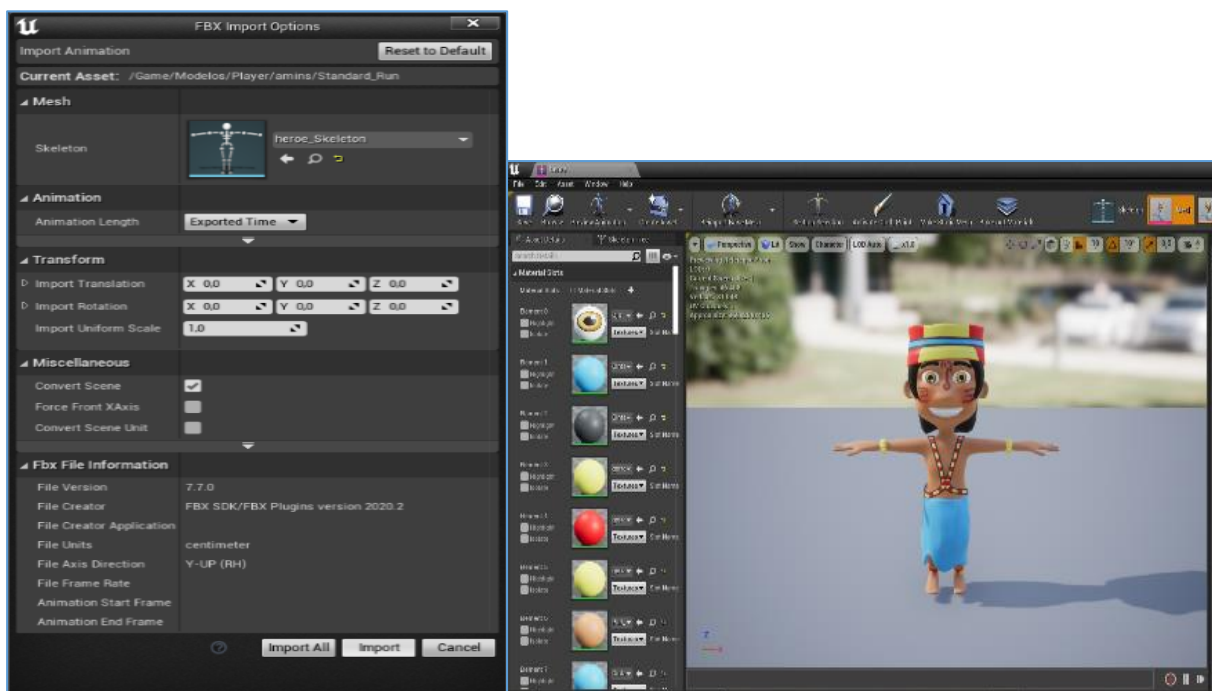
Figura 19. Flujo importación de modelos.



Fuente: Elaboración propia.

Primeramente, solo se exporta una vez el esqueleto del fbx como se muestra en la Figura 20:

Figura 20. Esqueleto FBX.



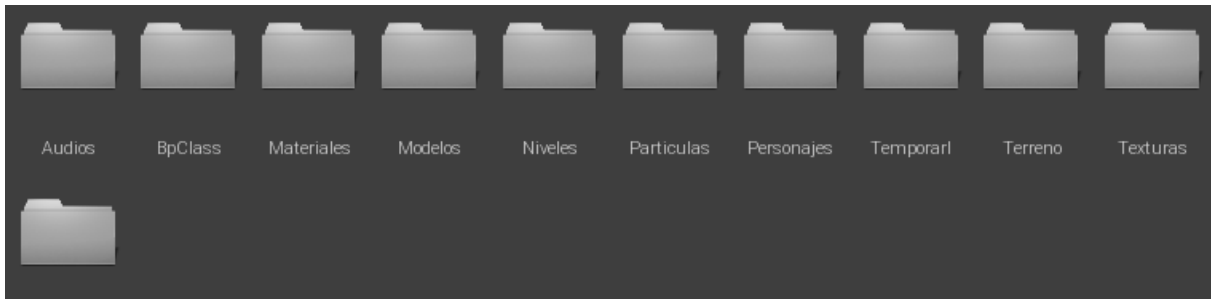
Fuente: Elaboración propia.

Nota: Las animaciones solo se importan sin las texturas y se selecciona el esqueleto ya existente de las misma dando clic en el botón import.

4.8 ESTRUCTURA DE CARPETAS DEL PROYECTO

Este proyecto se estructuró desde el framework de desarrollo las siguientes carpetas detalladas de la siguiente forma:

Figura 21. Estructura de carpetas de los Recursos empleados.

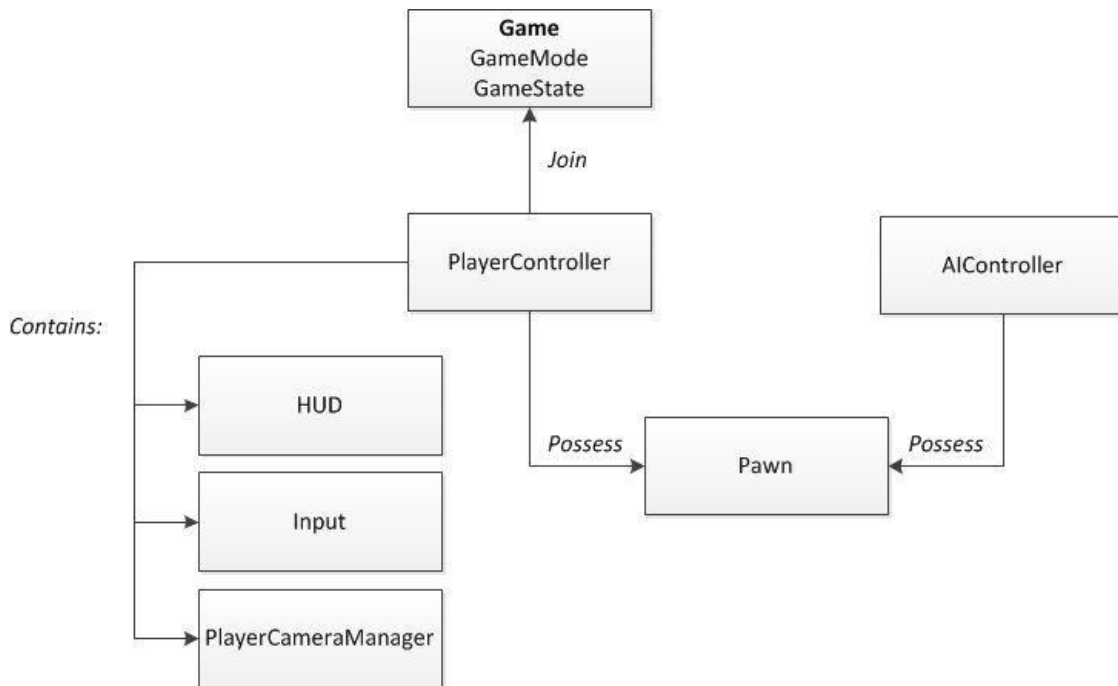


Fuente: Elaboración propia.

4.9 DIAGRAMA MODULAR (CAPA PRINCIPAL)

Al trabajar con metodología visual y utilizando Blueprints es muy importante identificar los Blueprints (BPs) que rigen el control general del juego, tal como se detalla en la Figura 22:

Figura 22. Framework Class Relationships.



Fuente: Página oficial de Unreal Engine.

En el próximo apartado se mostrará el diagrama modular empleado en el presente desarrollo.

4.10 IMPLEMENTACIÓN

El presente TFE, tiene un alcance limitado en la producción debido primero a una limitación temporal, una limitación de competencias y una limitación de recursos.

El TFE pretende ser un demostrador que permita visualizar una representación gráfica del texto aquí referido.

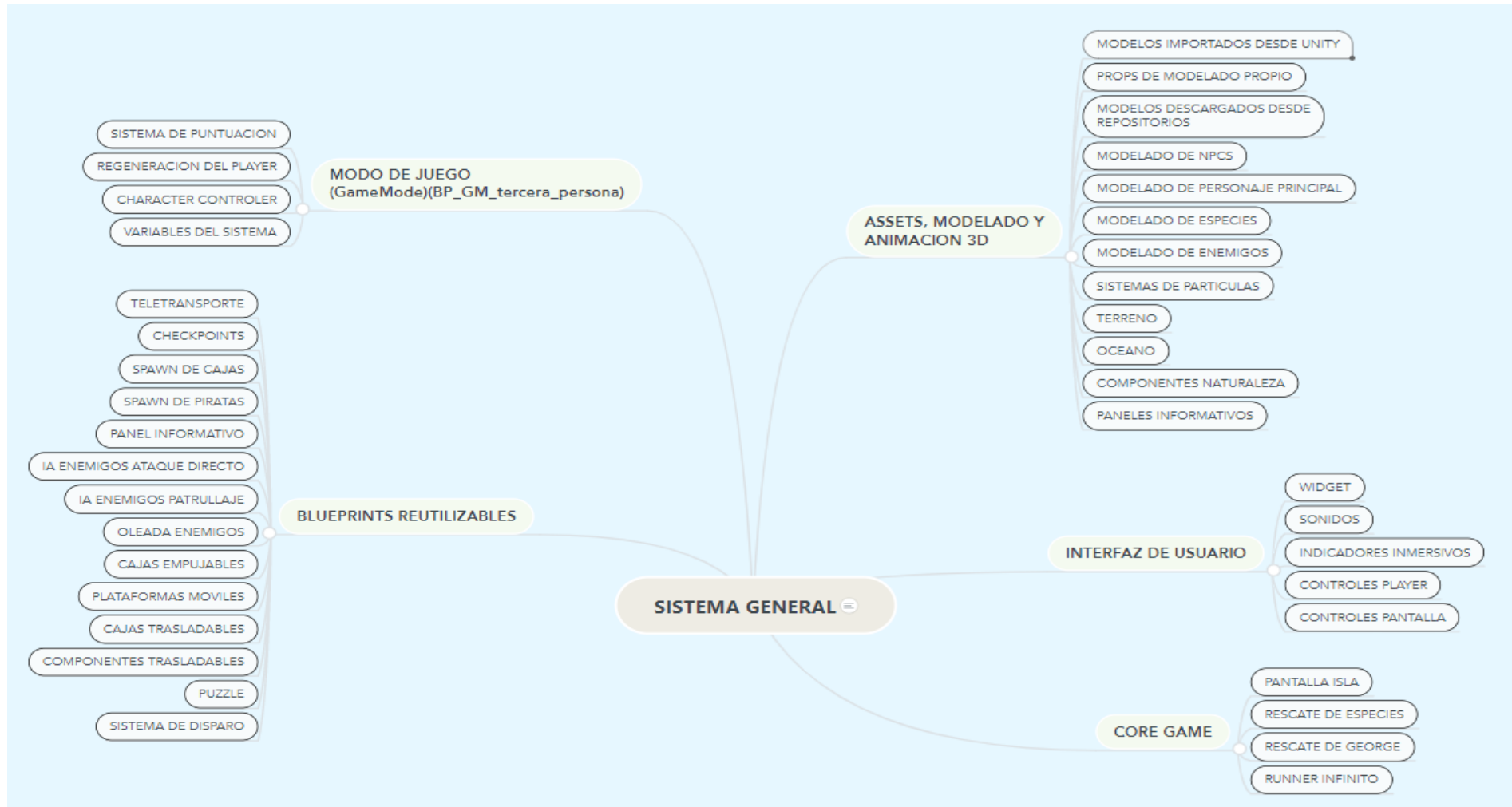
Se ha desarrollado un mapa mental que permite una mejor visualización de la implementación del presente trabajo. El enlace en MindMeister:

<https://mm.tt/2375108832?t=KwxxNQAIPr>

El sistema principal queda dividido en 5 secciones donde se ha focalizado el esfuerzo de implementación: assets y objetos 3d, blueprints generales reutilizables, interfaz de usuario, modo de juego y core game.

A continuación, en la Figura 23, puede visualizarse la propuesta de implementación:

Figura 23. Esquema general de implementación.



Fuente: Elaboración propia.

En la Figura 24, se presentan los diferentes esquemas que permiten ir al detalle en cada una de las partes:

Figura 24. Esquema implementación MODO DE JUEGO.

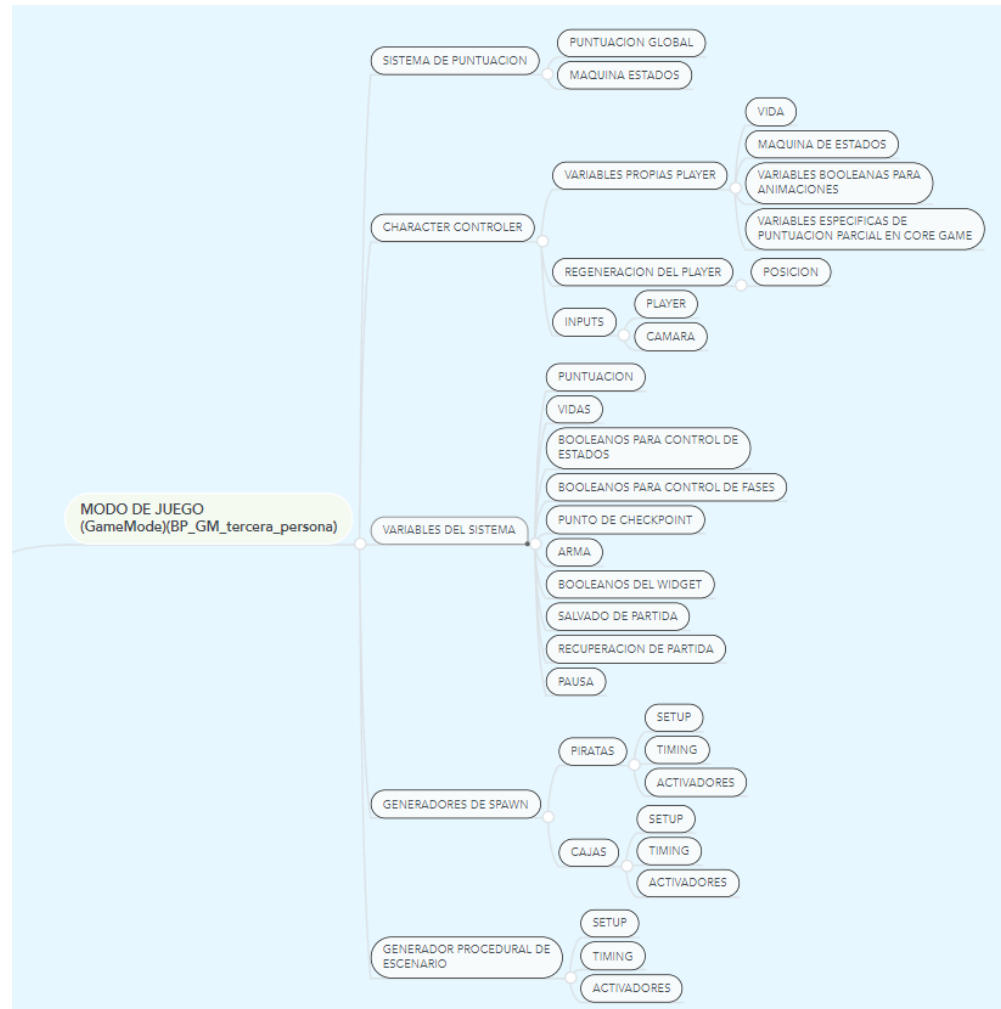
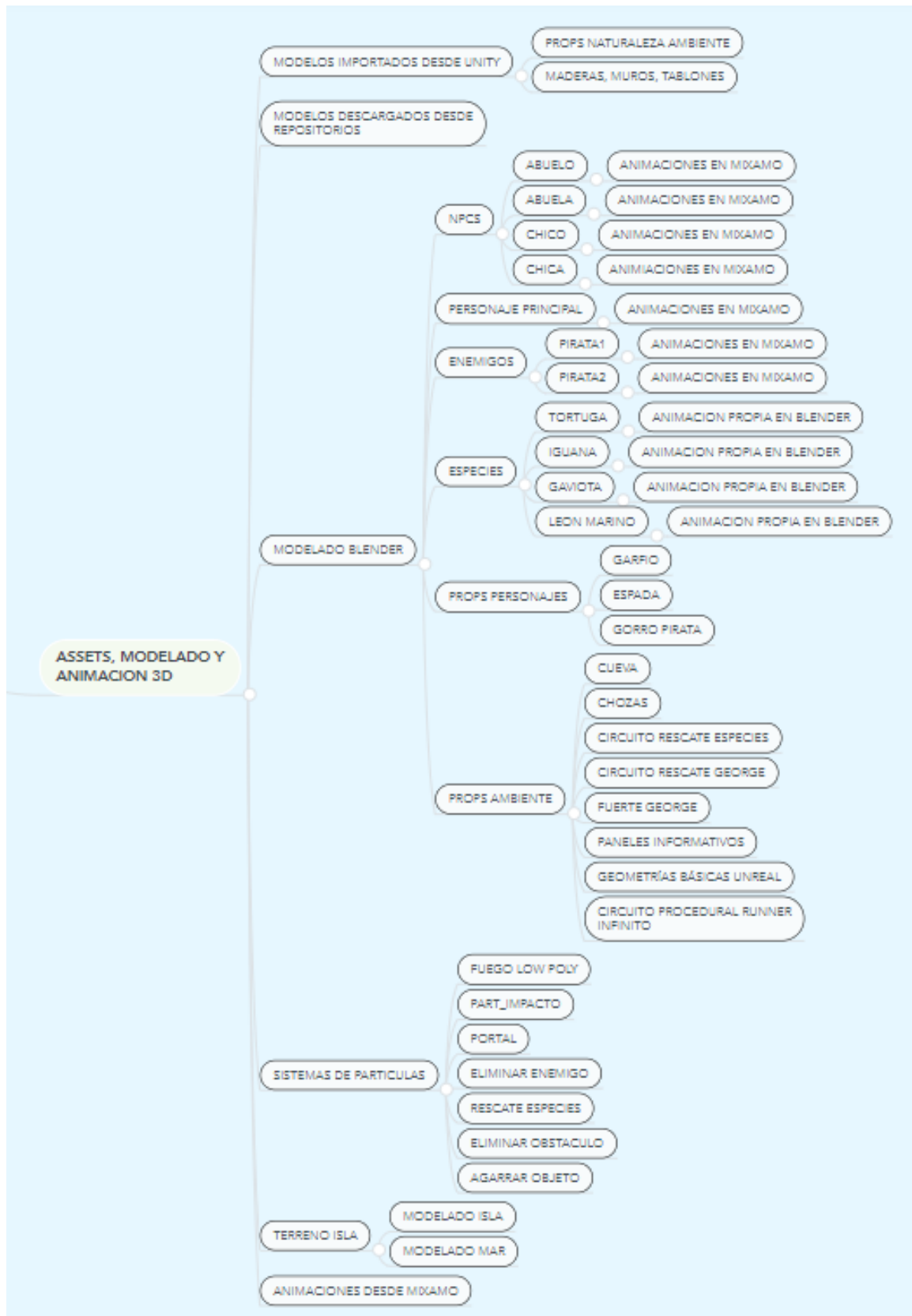


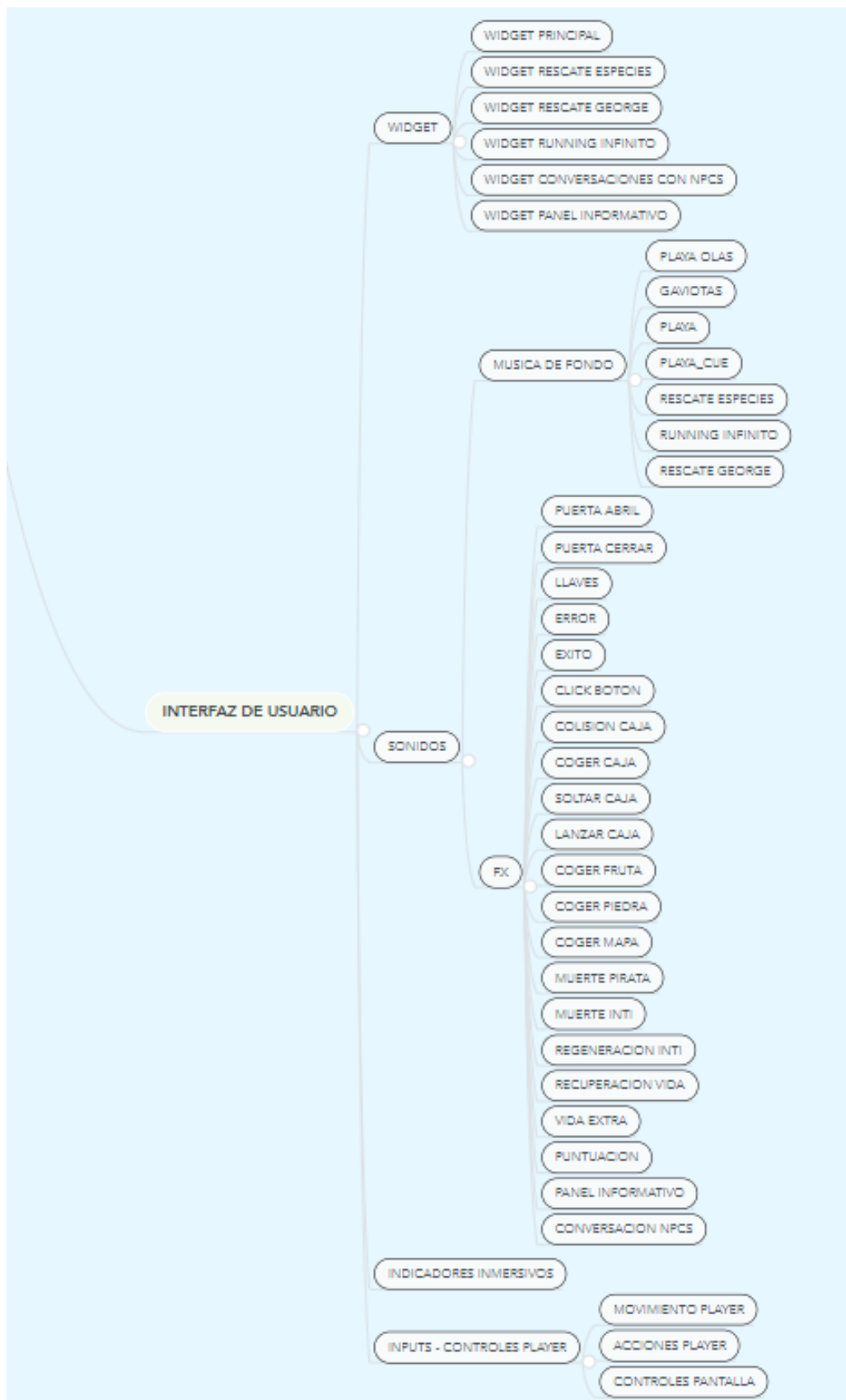
Figura 25. Esquema general de recursos y animaciones desarrollados e importados.



Para la implementación de recursos gráficos se han usado diferentes modalidades de obtención de recursos, desde la creación de recursos propios con Blender, a la importación desde el store de Unity, detallados en la Figura 25.

La interfaz de usuario engloba principalmente los recursos implementados para la comunicación de/con el usuario y para la generación de inmersividad en la experiencia de usuario. Considerando principalmente el HUD, los sonidos y los controles del usuario se detallan en la Figura 26:

Figura 26. Esquema general de implementación de la interfaz de usuario.



Finalmente, se detalla en la Figura 27, la implementación realizada (en el nivel 1 propuesto) para las misiones 1, 2 y 3, considerando también la implementación en la isla principal Isabella, vinculo de comunicación con las misiones desarrolladas (explicado con mayor detalle en el GDD, apartado diseño de niveles y en el GDD, apartado Gameplay).

Figura 27. Esquema general de implementación de la zona ISABELLA.

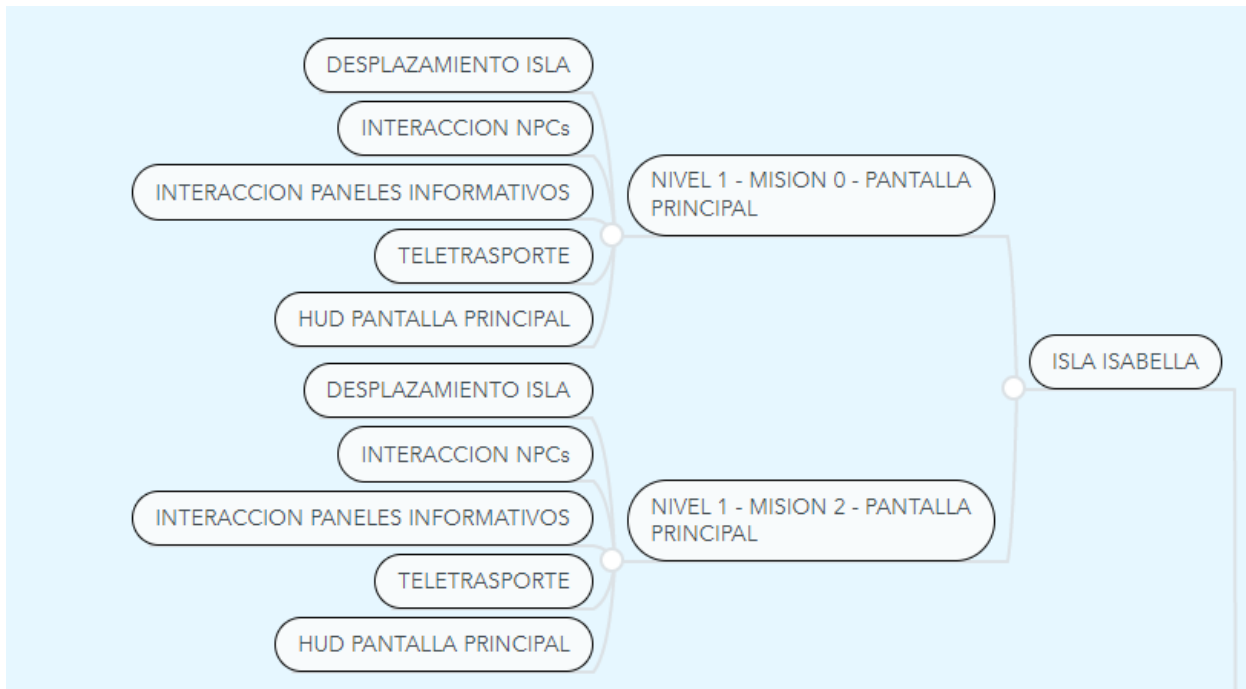


Figura 28. Esquema general de implementación de la MISION 1 – RESCATE DE ESPECIES.

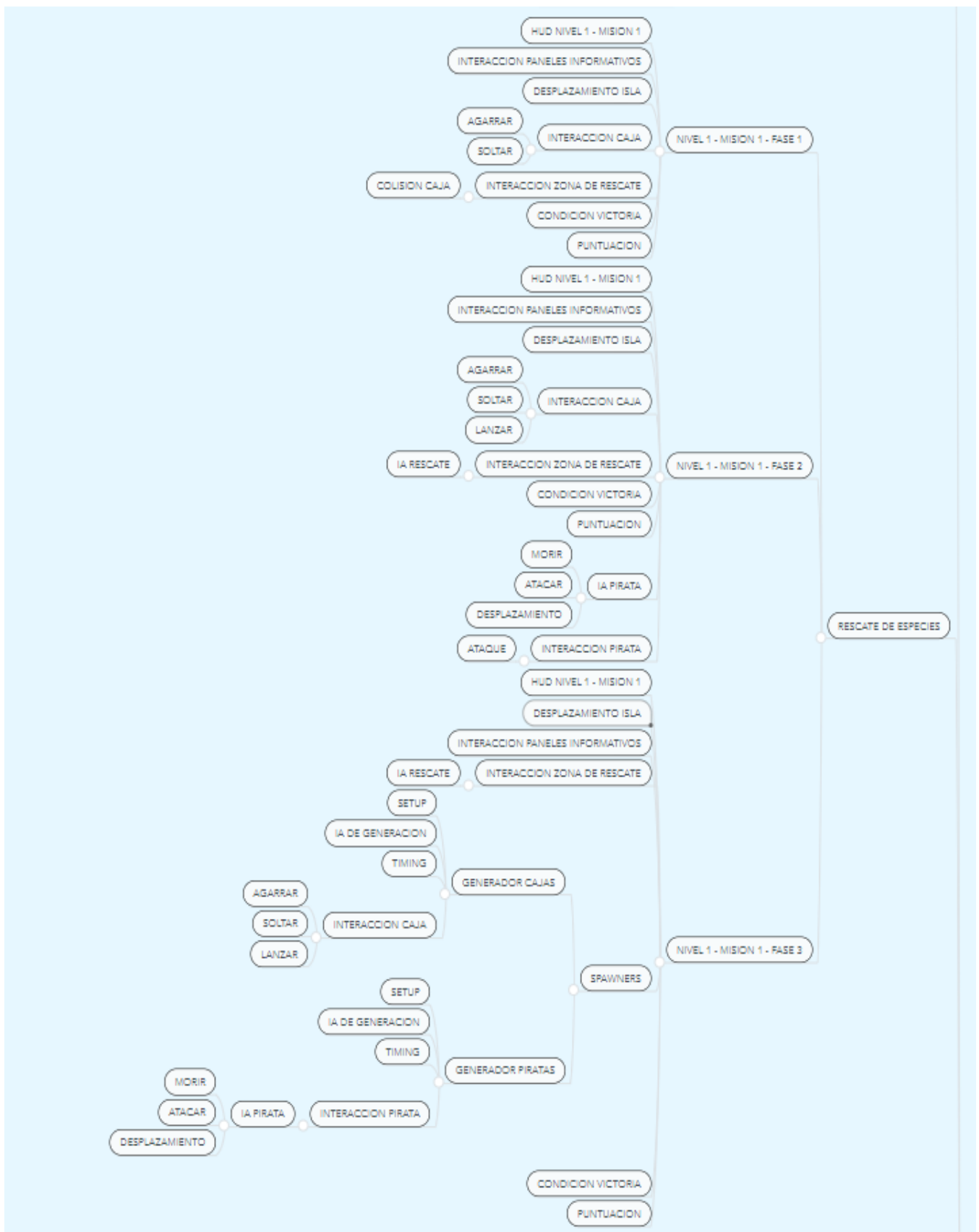


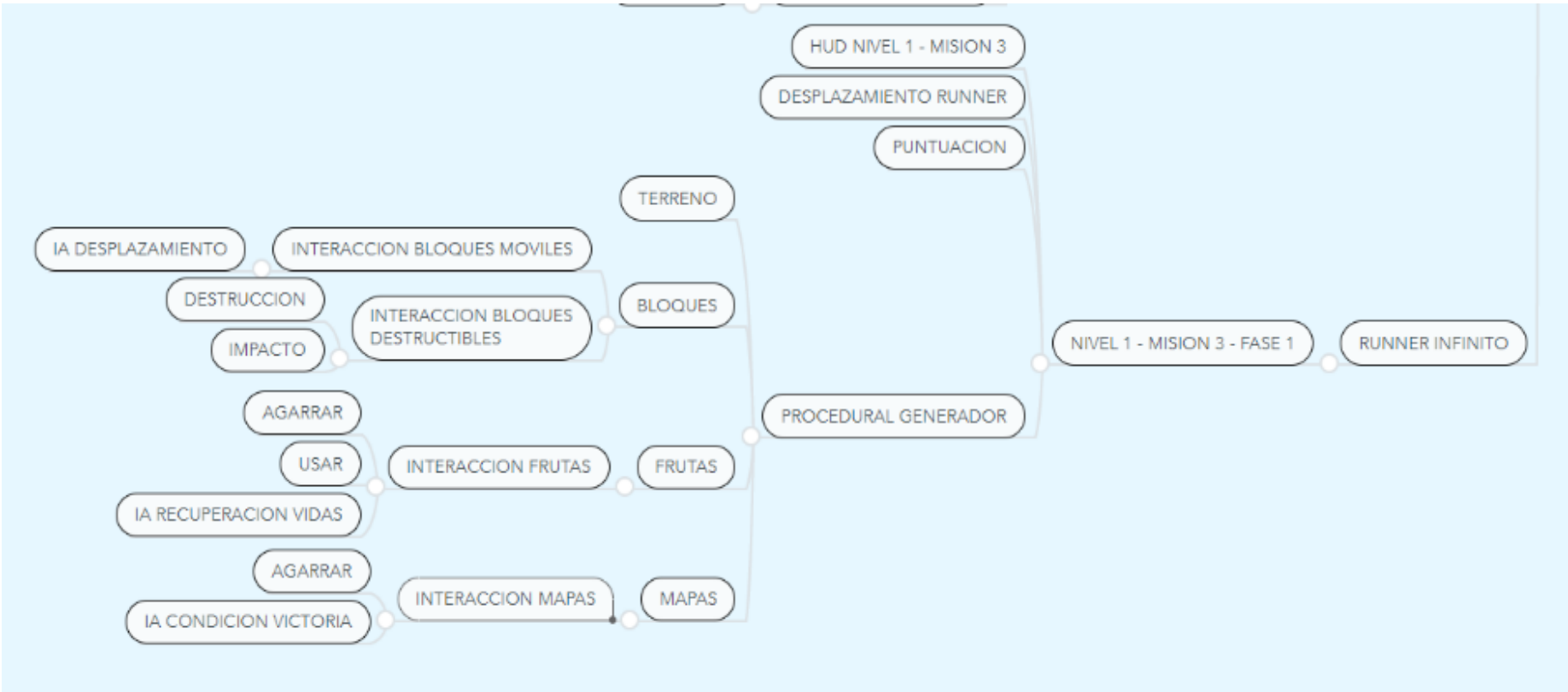
Figura 29. Esquema general de implementación de la MISION 2 – RESCATE DE GEORGE, fases 0 y 1.



Figura 30. Esquema general de implementación de la MISION 1 – RESCATE DE GEORGE fases 2, 3 y 4.



Figura 31. Esquema general de implementación de la MISION 3.

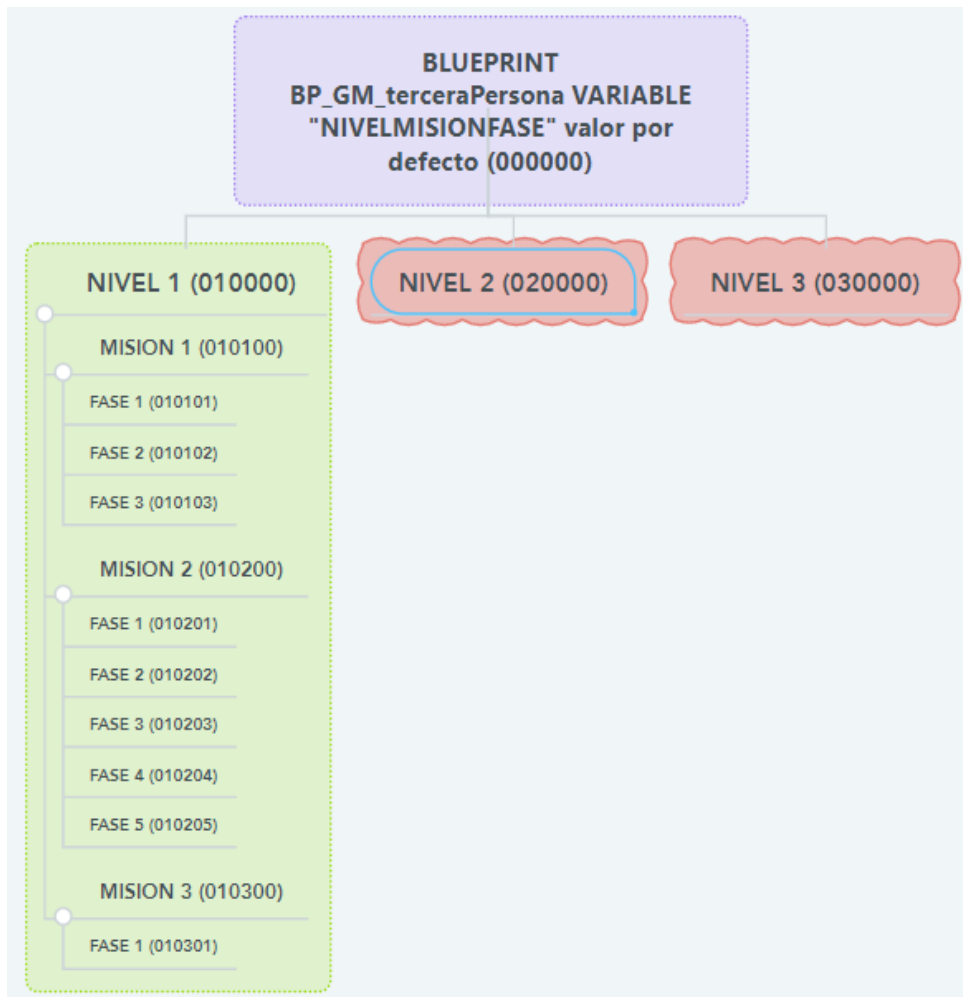


Así, para el correcto desarrollo del videojuego y una buena comunicación entre las diferentes clases (tal y como se muestra en el diagrama de flujo anterior, se han creado diferentes variables almacenadas tanto en la clase del Jugador principal "Inti" como en la clase que especifica el modo de juego.

❖ **Variables en la clase BP_GM_terceraPersona:**

Existen dos sistemas de puntuación, la primera es la puntuación general independientemente de las veces que han matado al personaje, variable Puntuación, que se almacena en la clase Modo de Juego para que no se pierda, podemos comprobar que también almacenamos el total de Vidas y un campo importante, Puntogegeneracion, es una variable que nos permite almacenar la posición del jugador en el momento en el que muere. En la Figura 32, se describe la variable NIVELMISIONFASE y su codificación:

Figura 32. Variable NIVELMISIONFASE y su codificación.



❖ **Variables en las diferentes clases Blueprints "BP" del (2 nivel):**

Las siguientes variables son necesarias para las estadísticas y el progreso del nivel, su creación y almacenamiento en los Blueprints del Modo de Juego, Jugador y del Widget son necesarias por lo cual, en la Tabla 5, Tabla 6 y Tabla 7, se describe el tipo y el motivo de cada una de ellas:

Tabla 5. Descripción de las variables en el BP_Inti.

BP_Inti		
Nombre	Tipo	Descripción
Intentos	Integer	El número de intentos que el jugador ha sido teleportado a un checkpoint, información consultada al término del nivel para las estadísticas del jugador.
EnemigosDerr otados	Integer	El número de enemigos abatidos, información que aporta para la elaboración de las estadísticas.

MuertesNivel	integer	Almacena el número de muertes del jugador en el nivel actual, su información es requerida para realizar las estadísticas al final de cada misión.
--------------	---------	---

Nota: Las variables aquí descritas se reinician su valor al término o salida del nivel de turno.

Fuente: Elaboración propia.

Tabla 6. Descripción de las variables en el BP_GM_terceraPersona.

BP_GM_terceraPersona		
Nombre	Tipo	Descripción
CheckPoint	array 3d	Guarda la posición y rotación del jugador, en caso que el jugador pierda, se teleporte o se reinicie el nivel se podrá consultar para posicionar al jugador en la posición del contenido de la variable.
Llaves	array	Guarda el id de las llaves conseguidas, la información será consultada al momento de intentar acceder a una puerta.
Puntuacion	Integer	Almacena el cálculo de ((EnemigosDerrotados * Vida)-MuertesNivel)-Intentos , información que se consultada al finalizar el nivel de manera exitosa.

Fuente: Elaboración propia.

Tabla 7. Descripción de las variables en el BP_Widget.

BP_Widget		
Nombre	Tipo	Descripción
Msjlmg	String	Almacena el nombre del material que se mostrará en la parte superior izquierda, se usa para la interfaz de diálogos.
BotonA	Boolean	Guarda cuando se presiona o suelta la tecla A , La información es imprescindible para dar feedback al usuario sobre la tecla ha pulsado.
BotonS	Boolean	Guarda cuando se presiona o suelta la tecla S . Igual que el botón A.
BotonD	Boolean	Guarda cuando se presiona o suelta la tecla D . Igual que el botón A.
BotonW	Boolean	Guarda cuando se presiona o suelta la tecla W . Igual que el botón A.
MouseL	Boolean	Guarda cuando se presiona o suelta el click izquierdo del mouse . Igual que el botón A.
MouseR	Boolean	Guarda cuando se presiona o suelta el click derecho del mouse . Igual que el botón A.

Fuente: Elaboración propia.

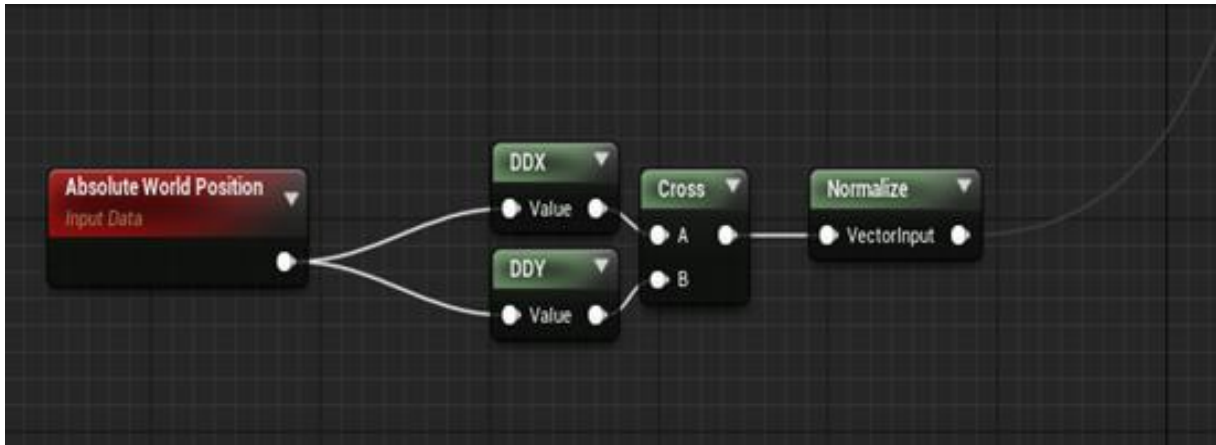
4.10.1 Implementación del Escenario Isla Isabela

- ❖ Creación de terrenos Lowpoly en Unreal 4. La implementación de la apariencia de terrenos LowPoly se ha creado un material Mat_low_suelo en el cual se ha realizado cálculos sobre la posición del mundo mediante los parámetros DDX y DDY una unión

mediante el nodo CORSS y la normalización del resultado asignado al normal del material del terreno, mostrados en la Figura 33:

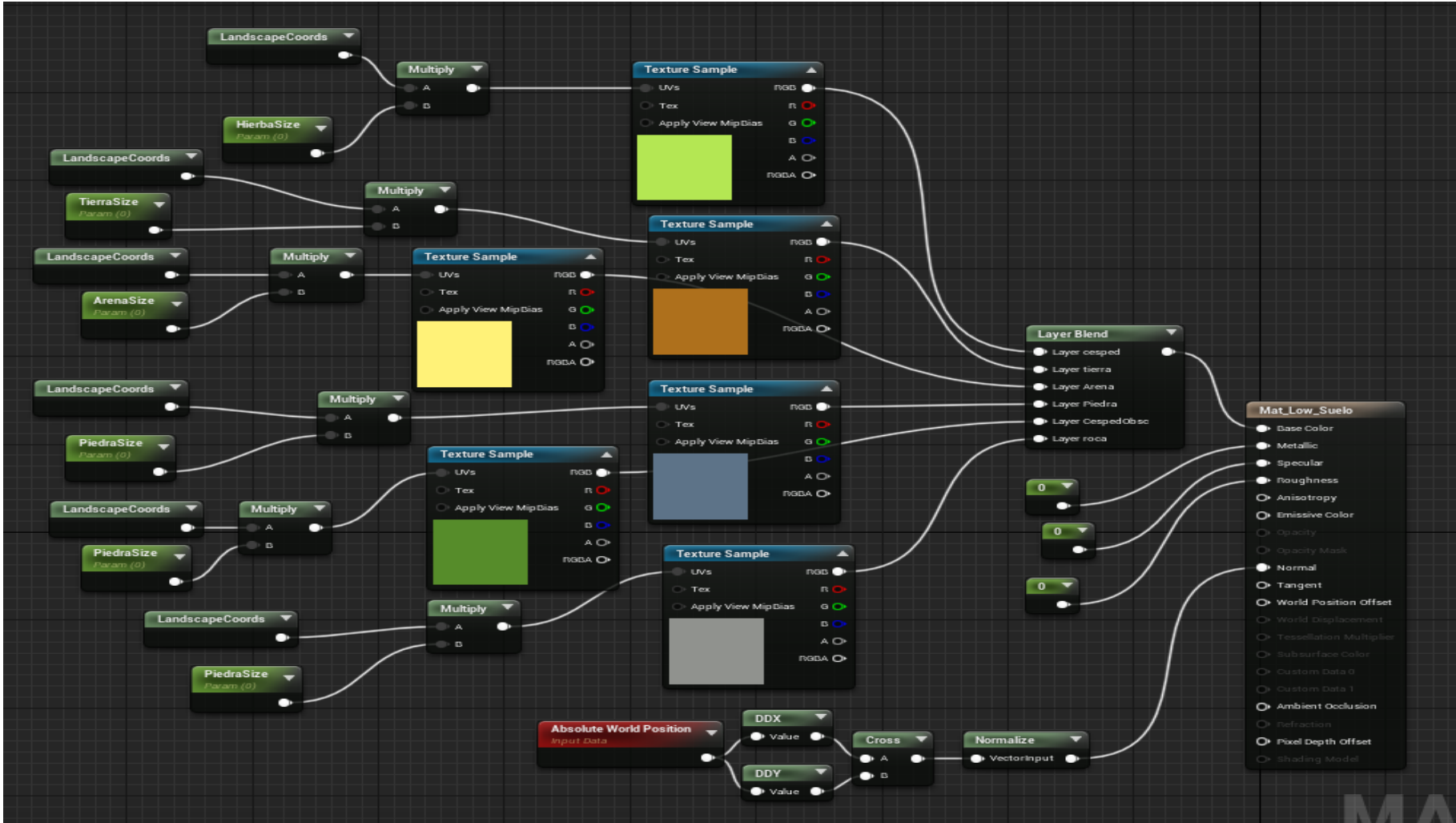
Para la texturización del terreno se ha multiplicado las coordenadas por 0 y asignado a las UVS, mostrado en la Figura 34:

Figura 33. Configuración del terreno para la Isla Isabela.



Fuente: Elaboración propia.

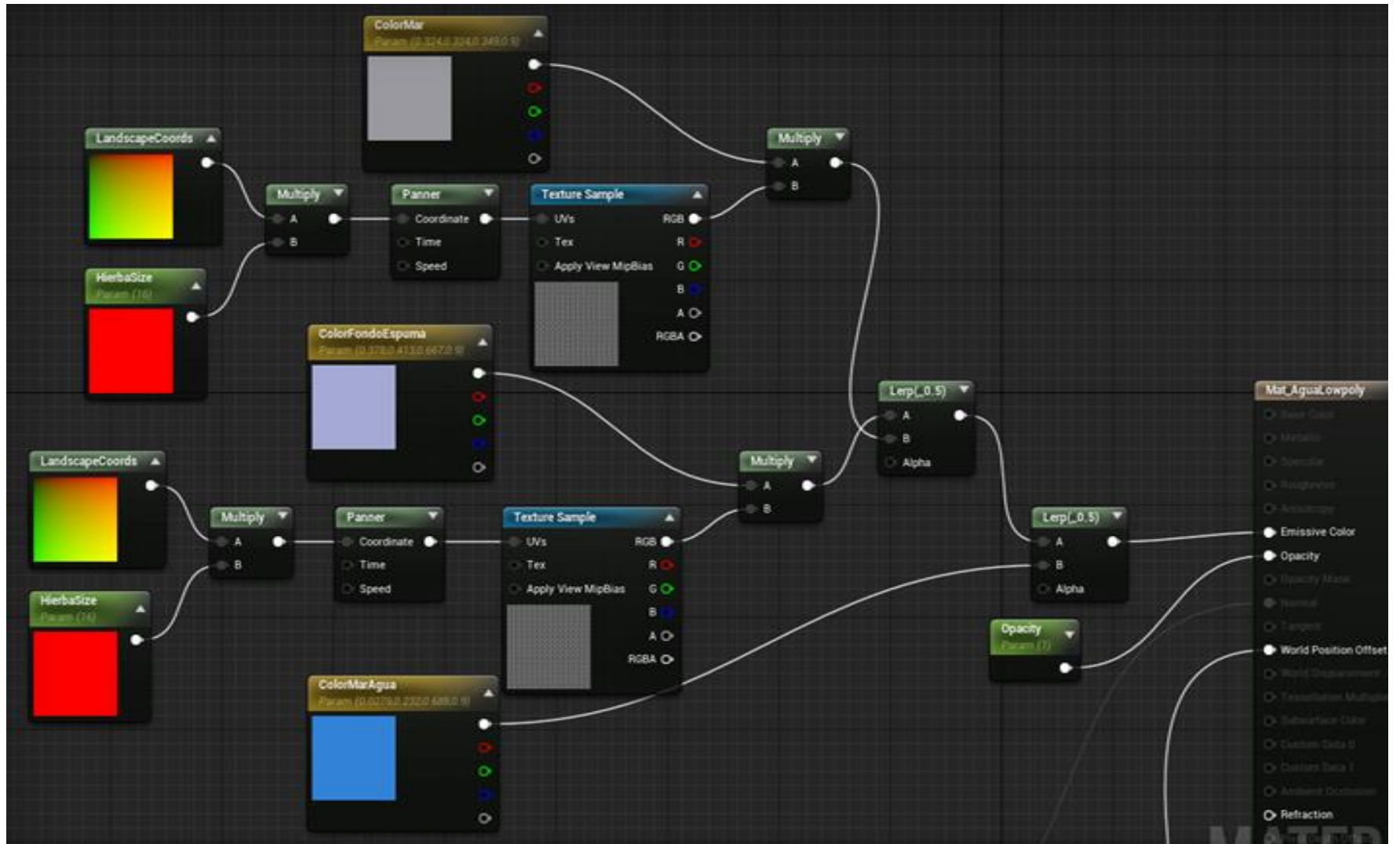
Figura 34. Multiplicación de las coordenadas por 0 y asignadas a las UVs.



❖ Creación de material dinámico del mar Mat_AguaLowpoly.

Para la simulación de mar se ha usado una textura y se ha duplicado la misma para lograr esto se ha usado un material dinámico. Mediante la multiplicación de landscapecoords por una constante el resultado se asigna a un nodo PANNER y asignado al UVS de un texture sample (Voronoi) multiplicado y el resultado se le asigna a un LERP o interpolación. este proceso se repite para los dos primeros colores ya sea del color de la espuma como del mar, ambos resultados se interpolan con un tercer color (Color del mar más claro), el resultado de esta interpolación se asigna al **emissive color** del material. Es importante manejar la opacidad de la tercera textura mediante una variable, se puede observar esta combinación en la Figura 35:

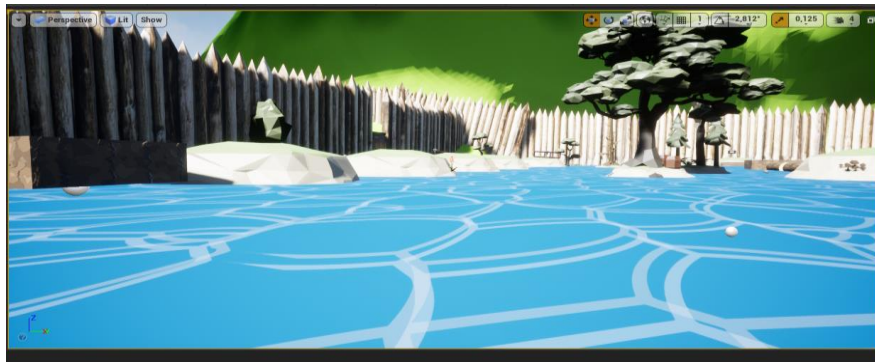
Figura 35. Creación del material dinámico del mar.



❖ Aspecto Lowpoly del mar.

Para cambiar la apariencia al material del agua a bajo poligonaje, es necesario agregar los siguientes nodos. Absolute World Position del cual agregaremos los nodos DDX y DDY a los que se asignan a un nodo CROSS y su resultado se asigna al nodo NORMALIZE y finalmente el resultado de toda la operación se asigna al normal del material de agua, como se puede observar en la Figura 36:

Figura 36. Aspecto Lowpoly del mar.

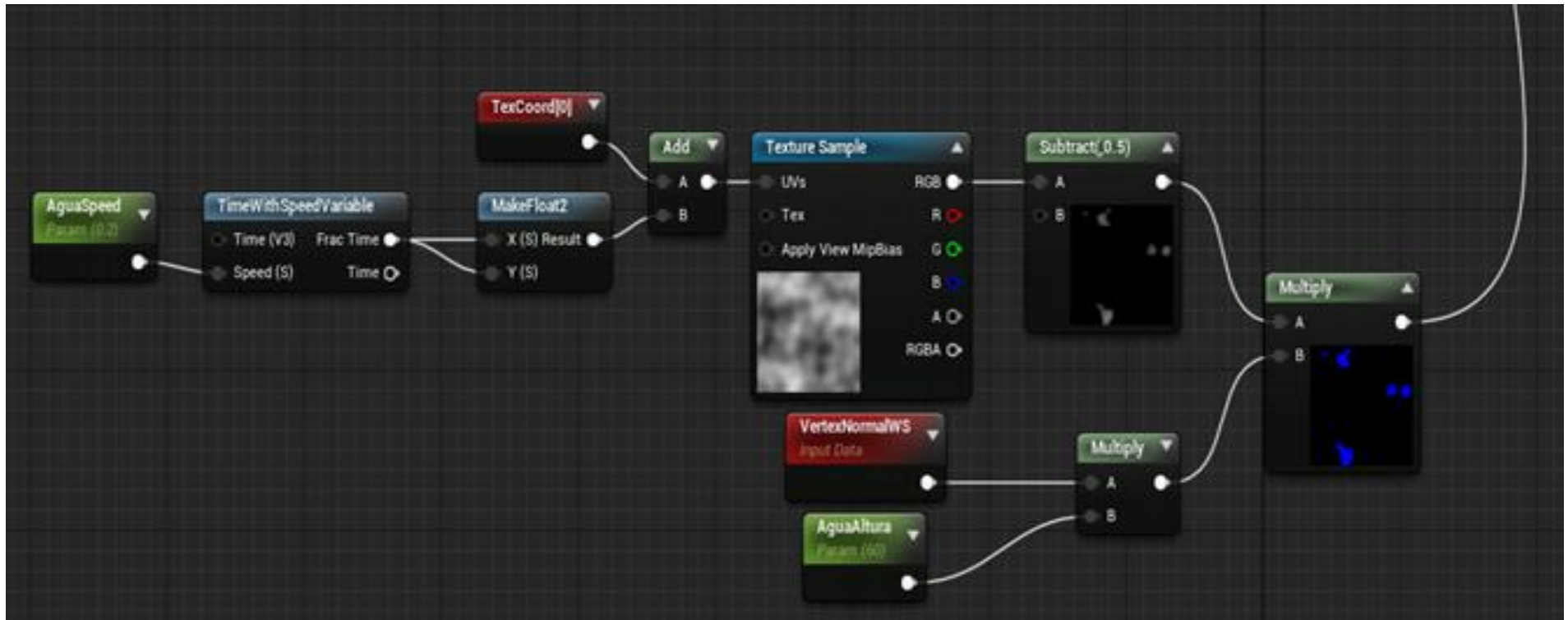


Fuente: Elaboración propia.

❖ Movimiento del Agua del Mar.

El movimiento de olas de mar se consigue agregando una constante con el cual se manipulará la velocidad en el que se desplazan las texturas, un nodo TimeWhitSpeedVariable al cual se agrega el nodo MAKEFLOAT2, con el nodo ADD le sumamos la coordenada y el resultado se agrega a un Texture Sample, al cual se usa un patrón Voronoi y se sustrae el RGB del mismo para ser multiplicado al resultado de esta operación se multiplica por una constante (Altura de las olas) por un VertexNormalws. El resultado obtenido se le asigna al material en el World Position Offset, como se muestra en la Figura 37:

Figura 37. Movimiento del agua del mar.



4.10.2 Implementación de mecánicas básicas de INTI

La configuración de los inputs descritos en la Tabla 8:

Tabla 8. Inputs del jugador.

Tecla del computador	Acción a realizar
W, Flecha Arriba	Movimiento adelante
S, Flecha Abajo	Movimiento atrás
A, Flecha Izquierda	Movimiento a la izquierda / giro izquierdo
D, Flecha Derecha	Movimiento a la derecha / giro derecho
Barra Espaciadora	Saltar
Q	Ver inventario
Botón Derecho del Ratón	Agarrar / Soltar / interactuar con objeto
Shift	Caminar despacio en runner / Correr
E	Usar Objetos desde inventario

Fuente: Elaboración propia.

❖ BP (BluePrints) del movimiento del player

Se ha implementado una variable en el BP del Modo De Juego que nos permite registrar el nivel, la misión y la fase en la que se encuentra el jugador. La variable se llama NivelMisionFase y se codifica con dos dígitos para cada uno de estos datos, por ejemplo, en el Nivel 1, Misión 1 Fase 2 se identifica como 010102.

Esto es especialmente crítico para el Nivel1, Misión 3 donde se deben cambiar las mecánicas del player con respecto a las misiones 1 y 2.

En las misiones 1 y 2 el jugador se mueve usando los controles habituales WASD y las flechas. En la misión 3, es un Runner Infinito, el jugador por defecto corre, y se usan los controles de dirección para un cambio de dirección.

- ❖ **Nota:** Con la tecla shift se altera el movimiento del player a 600 permitiendo que la animación de correr se dispare y aumentando la velocidad del mismo.
- ❖ Blendspace de las animaciones.

Es importante tener en cuenta que las animaciones listas para arrastrarlas y dependiendo de la velocidad del player las mismas pasarán de un estado a otro (Idle, caminar y correr).

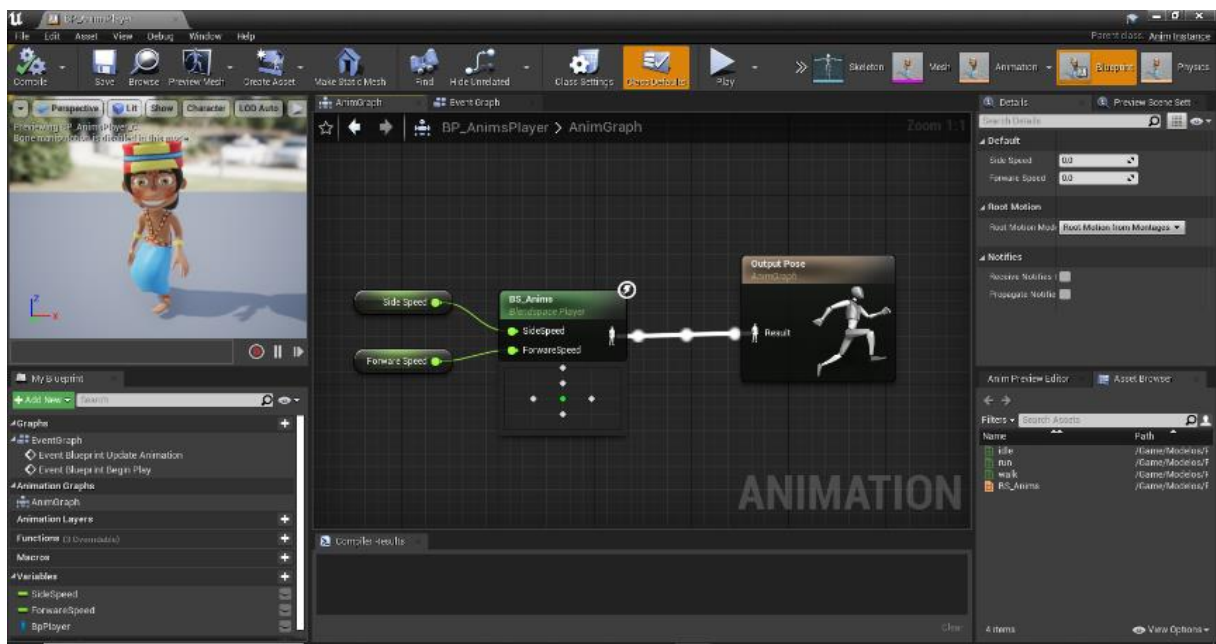
- ❖ BP de las animaciones (Event graph)

Es importante validar que exista un player controller para poder obtener la información de la velocidad y dado que se usa un evento que se ejecuta a cada frame, se debe validar la existencia de un character controller.

- ❖ BP de las animaciones (Anim Graph)

Se usaron una máquina de estados en la cual el default contiene un mapa Blendspace de movimiento, como se muestra en la Figura 38:

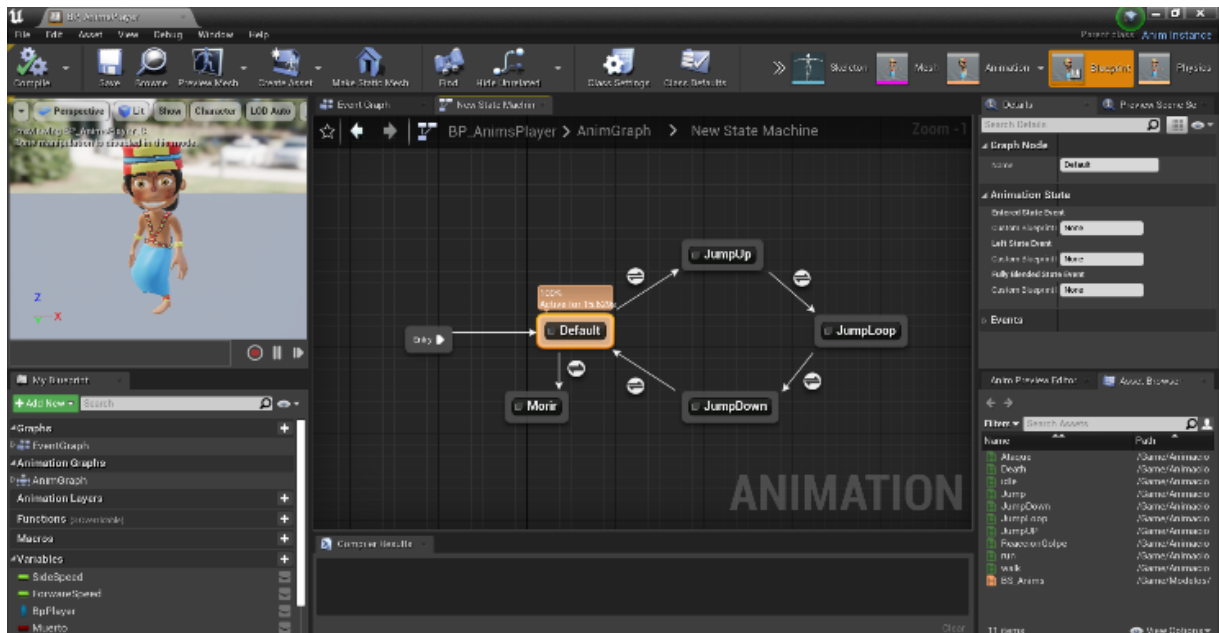
Figura 38. BP animaciones (Anim Graph).



Fuente: Elaboración propia.

Se define mediante variables las entradas de velocidad permitiendo el cambio de animaciones, como se puede observar en la Figura 39:

Figura 39. Máquina de estados de la animación Inti.

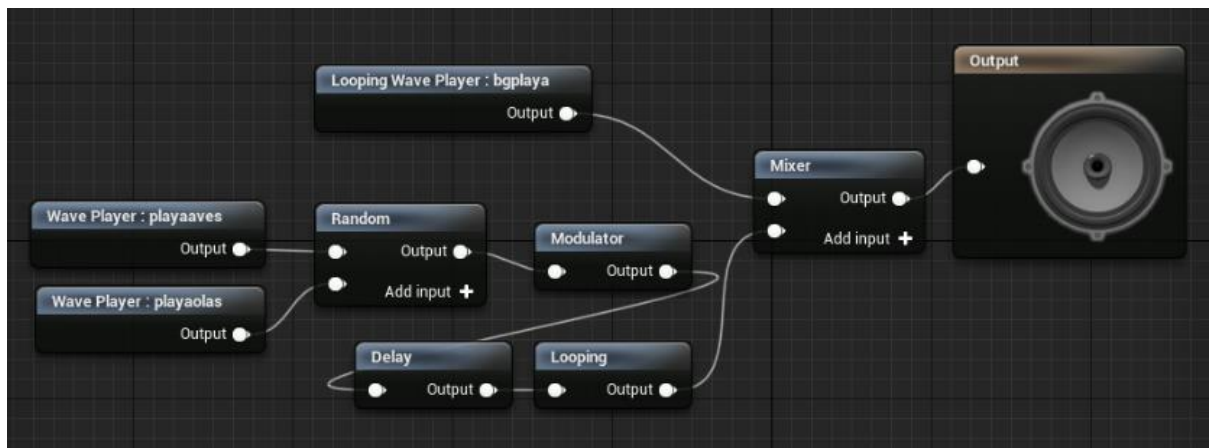


Fuente: Elaboración propia.

4.10.3 Implementación del sistema de Audio

Con el componente audio que se ha creado el sonido del entorno y un mix entre 3 sonidos al azar asemejando a la playa, sus olas y las aves a su alrededor, explicados de mejor manera en la Figura 40:

Figura 40. Componente de sonido.



Fuente: Elaboración propia.

4.10.4 Implementación del sistema de recogida de objetos

La implementación del sistema de interacción con objetos queda generalmente resuelta mediante la detección de colisiones entre el player y el objeto, en ocasiones también reforzado con la necesidad de disponer de una tecla pulsada en el momento de la colisión:

Coger objetos: Mediante colisiones se detecta al jugador y se presentan las ayudas visuales para incitar al jugador a interactuar con el objeto.

Soltar objetos: La acción de soltar objeto simplemente consiste en liberar el objeto del socket.

Lanzar objetos: La acción de lanzar el objeto realiza dos tareas, liberar el objeto del socket a la vez que aplica una fuerza sobre el mismo utilizando el objeto como arma (en el caso de las cajas principalmente)

Empujar objetos: Algunos objetos como columnas o cajas mediante la detección de colisión permite que el jugador aplique una fuerza para empujarlos.

Postes: Tableros de información que guían al jugador ya sea con pistas o consejos.

Las puertas: Puertas tanto con llaves como con interruptores dan pistas al jugador en caso de estar cerradas.

4.10.5 Implementación del sistema de SPAWN

El sistema de implementación de SPAWN desarrollado tanto para generar piratas como para generación de cajas tienen un elemento común y es que se encuentran limitados por una variable del sistema que bloquea la generación de elementos adicionales en base a los elementos que se encuentran activos en ese momento (en el caso de los piratas, no puede haber más de 18 piratas en la escena) o en el caso de las cajas, no se pueden generar más de 5 cajas de cada especie (ya que son indestructibles).

El sistema comienza con un CAST al Modo De Juego para tener acceso a esas variables.

En el caso del sistema de SPAWN de cajas genera cajas aleatorias, para ello, un sistema Random genera un INT entre 3 posibles. El switch nos permite llamar al SpawnActor correspondiente a la caja de cada una de las especies en función del INT obtenido en el Random. Para controlar el número de cajas, en la parte final se incrementa en 1 el contador correspondiente a la caja Spawneada.

La localización de la nueva caja creada se calcula con un Random Point In Bounding Box alimentado por un Get World Location como origen y un Get Scaled Box Extend del elemento AreaGeneracionCubos.

En el caso del SPAWN de Piratas el algoritmo es más simple pues evitando el sistema Random y Switch directamente se controla el número de piratas activos y en caso de ser menor de 18 invocamos la función SpawnAI From Class, donde la clase es el BP del Enemigo.

La localización del nuevo pirata se calcula con un Random Point In Bounding Box alimentado por un Get World Location como origen y Un Get Scaled Box Extend del elemento Area Generacion Piratas.

4.10.6 Implementación del sistema de resolución de enigmas

ACERTIJOS

Mediante colisiones con el player se guarda la secuencia de cada colisión y se compara con la secuencia previamente almacenada, dando un feedback visual y sonoro ya sea por aciertos como por errores, una vez completado el acertijo se abrirá automáticamente la puerta de turno.

Mediante un Class Blueprint se definen en 4 box y cubos los cuales se les agrega un material Matyellow (amarillo) por defecto y que se va cambiado dependiendo si la secuencia es acertada, este se reemplaza por el MatOk (Color verde) y Matbad para los intentos fallidos. También se dispone de sonidos que según la secuencia estos son clic y error.

PUERTAS

Puertas con llave: al detectar la colisión con el player la puerta consulta si este posea la llave requerida para que la puerta se abra y sino presentará un mensaje solicitando la llave o la ubicación de donde puede encontrarla. Puerta con botón: al igual que la anterior solicitará al jugador que deba pulsar un botón para poder dar acceso al jugador.

4.10.7 Implementación del sistema de Regeneración del personaje cuando muere y guardado de CheckPoint.

El sistema de Regeneración del personaje depende de diferentes módulos de implementación.

Por otro lado, el blueprint BP INTI dispone de una acción que continuamente actualiza la posición de regeneración garantizando que el jugador al regenerarse recupera la posición en la que se encontraba al morir, esto se hace almacenando Get Actor Transform del jugador sobre la variable PUNTOGEGENERACION del tipo Transform que se encuentra en El Modo De Juego. Esta acción se llama continuamente para garantizar que actualizamos la posición del player.

En el Modo De Juego, el módulo de Generacion, lo primero que debe realizar es la destrucción de todos los elementos SPAWN generados durante la vida anterior. Para ello usamos un Get All Actors From Class (para cada una de las clases Instanciadas Caja1, Caja2, Caja3 Y Pirata) recorriendo el Array mediante un For Each Loop y terminando con un Destroy Actor.

Seguidamente el sistema debe comprobar si el jugador tiene vidas suficientes para volver a regenerarse y continuar la partida. En caso de que así sea se hace una llamada a SpawnActor BP INTI (player) en terminando con un Possess dándole el control Get Player Controller al nuevo BP INTI.

4.10.8 Implementación de la IA de los enemigos

Los enemigos se generan cuando el player se encuentra dentro de un radio específico por ende intervienen dos BluePrints Bp_enemigos y Bp_oleadasenemigos:

- ★ Bp_Enemigos: Para la inteligencia de los enemigos se han implementado mediante el nodo AI_MOVE_TO con un sistema de patrullaje en un rango determinado, con puntos aleatorios de destino y periodos de tiempo donde el enemigo se detiene simulando descanso, mediante los métodos de detección disponibles del motor se altera la velocidad para alcanzar al player y posteriormente reducir su vida en caso de alcanzarlo, sus variables están detalladas en la Tabla 9:

Tabla 9. Variables BP_Enemigo.

BP_Enemigo		
Nombre	Tipo	Descripción
Radiopatrulla	Float	Define el rango de patrullaje del enemigo
Perseguir	Boolean	Estado que se activa si ha sido atacado o ha visto al player.
Puedemoverse	Boolean	Desactiva el patrullaje, útil cuando está en combate
velocidadBase	Integer	Define el valor base de movimiento "300"
Danio	Integer	Define el daño que causan sus ataques al jugador

Vida	integer	Almacena la cantidad de vida
Jugador	ref Player	Almacena la referencia del jugador

Fuente: Elaboración propia.

- ★ Bp_oleadasenemigos: Los enemigos se generan usando el nodo Spawn Actor, por medio de variables se pueden configurar los números de oleadas a y la cantidad de enemigos por oleadas, con los targets como puntos de SPAWN colocados de manera manual en el mapa se crea aleatoriedad al momento de generar enemigos, sus variables están detalladas en la Tabla 10:

Tabla 10. Variables del Bp_oleadasenemigos.

BP_Oleadasenemigos		
Nombre	Tipo	Descripción
Totaloleadas	Integer	Número total de las oleadas de enemigos.
EnemigosxOleadas	Integer	Cantidad de enemigos generados por oleada.
Intervalo	Float	el tiempo de espera para generar oleadas de enemigos.
Punto1-4	Target	Las posiciones en el mapa donde van a aparecer los enemigos
LastSpanw	Integer	Almacena el último punto donde se generó el enemigo
NewSpan	Integer	Almacena el nuevo punto para evitar que se repita.
Total	Integer	Números de enemigos a generar
Iniciar	Boolean	Almacena si se puede empezar a generar oleadas

Fuente: Elaboración propia.

4.10.9 Implementación del sistema de teletransporte

Existen dos sistemas de teletransporte del personaje, el Teleport como lanzadera al cambio de escenario y las Plataformas Móviles que desplazan al personaje dentro del mismo escenario:

TELEPORT

Se utiliza para devolver al jugador a un sitio seguro en caso de caerse del mapa o zonas no permitidas. El Checkpoint es un complemento que permite almacenar una posición segura en caso de ejecutar un Teleport no perder el avance.

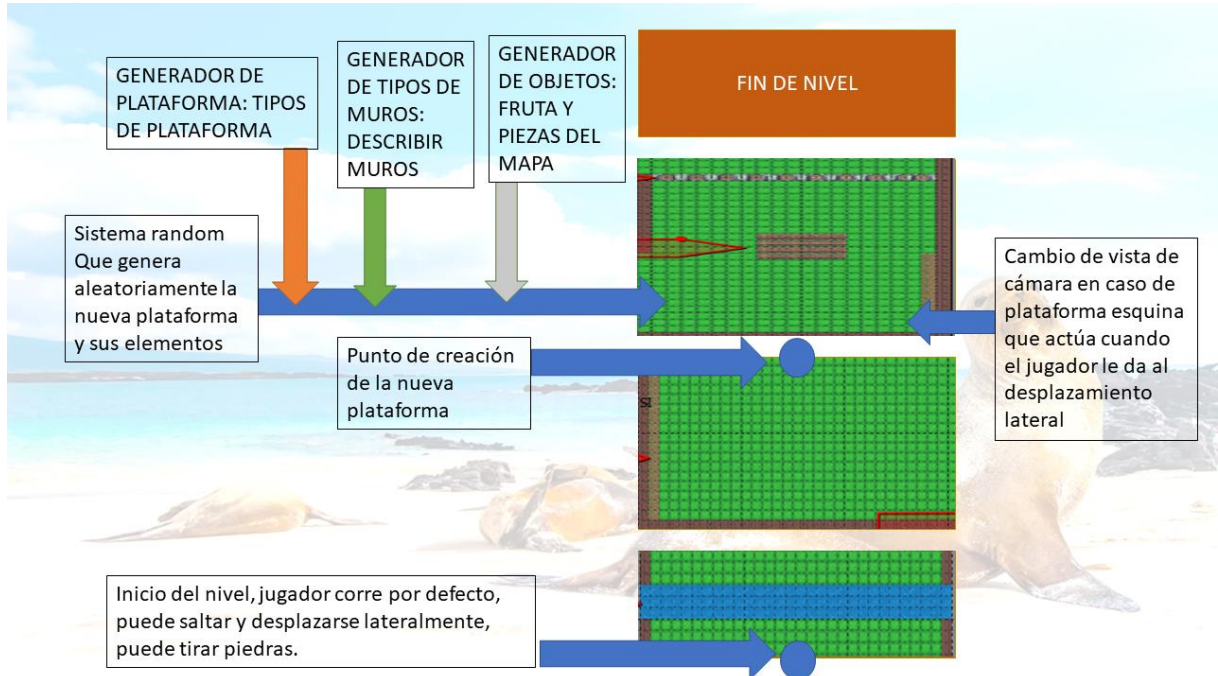
PLATAFORMAS

Mediante Timelaps controlamos la velocidad y con variables tanto de inicio como de destino, la plataforma cumple su recorrido, a la velocidad determinada por una tercera variable.

4.10.10 Implementación del mapa dinámico

Para la implementación del mapa dinámico se realizó un diagrama mostrado en la Figura 41:

Figura 41. Implementación del mapa dinámico.



Fuente: Elaboración propia.

Parte de un inicio de nivel, donde se ejecuta en cada Event Tick una función llamada Crear Plataforma, la cual por cada intervalo de tiempo de 1.4 segundos crea una plataforma aleatoria eligiendo entre 6 tipos diferentes de plataformas, partiendo de una PlataformaPadre, la función crear plataforma tiene la misma estructura para las diferentes 6 tipos de plataformas existentes, usando un SpawnActor From Class seleccionamos entre los diferentes actores existentes de plataforma que son: PlataformaRecta, PlataformaHueco, Esquina Izquierda, Esquina Derecha, Arriba Y, por último, Abajo.

Es importante aquí obtener el último punto de la plataforma creada, que será el nuevo punto de creación de la variable SiguientePlataforma, y así se construye sucesivamente el mapa dinámico.

Además, posee un componente EndOverlap, que hace que las plataformas al momento de esperar 2 segundos, se destruyan y así el camino dinámico no llegue a chocarse con plataformas creadas anteriormente.

En la plataforma padre, posee además la generación de dos objetos que son las frutas y las piezas del mapa, estas poseen un movimiento constante con la función Timelapse, lo que hace que las monedas y las piezas del mapa no estén estáticos. Es importante que al crear estos actores se almacenen en un Array de elementos para que a posterior, pasado un tiempo se eliminen para ahorrar recursos al momento de ejecutar el juego.

En dicha plataforma, se hace el uso de la opción Set Visible, para mostrar u ocultar obstáculos presentes en las plataformas, estos obstáculos aparecen de forma aleatoria, como son los muros indestructibles, los destructibles, los móviles, la puerta y unos obstáculos saltables.

Esta generación de obstáculos se lo realiza de manera aleatoria activando y desactivando esta propiedad SetVisible de cada uno de los obstáculos en la sección Details de cada componente.

4.10.11 Implementación del sistema de puntuación

El módulo de implementación de puntuaciones es un sistema implementado en las 3 misiones de forma diferente.

Para el sistema de puntuación se ha implementado una función Get_Puntuación_Text que nos da el valor de la puntuación del player y se lo envía en la UI, de igual manera para el sistema de las vidas. Cabe recalcar la programación realizada en este punto para este sistema se debe crear en el personaje y en la UI los siguientes BP.

Para la pantalla final se aprovecha el evento morir para continuar con esta parte, hacerlo visible y poner un botón que mediante la aparición del mouse se puede volver a jugar desde el inicio del juego.

A continuación, se describe cual es la lógica para la implementación del sistema de puntuación:

Misión 1, el objetivo es rescatar especies, en concreto 5 de cada una de las presentadas (iguanas, leones y tortugas), existe una variable temporal (cuanto vas a tardar en conseguirlo) y una variable vida (cuantas vidas tienes o cuanta vida te queda cuando finalices), también existe otra variable que es los Enemigos, cada Enemigo puede ser destruido o eliminado lanzando una caja.

CADA CAJA DE LEONES SON 5 PUNTOS

CADA CAJA DE IGUANAS SON 10 PUNTOS

CADA CAJA DE TORTUGAS SON 15 PUNTOS

CADA ENEMIGO ELIMINADO SON 20 PUNTOS

El sistema de puntuación se calcula al final de la misión con la siguiente formula:

$$\text{PUNTUACION} = (\text{CAJA 1} + \text{CAJA 2} + \text{CAJA 3} + \text{ENEMIGOS}) \times \text{VIDAS} \times 100 / \text{TIEMPO}$$

MISION 2, el objetivo es rescatar a la tortuga George, el sistema de puntuación en este nivel se implementa de la siguiente forma:

$$\text{PUNTUACION} = \text{ENEMIGOS DERROTADOS} \times \text{VIDA} \times 100 / \text{TIEMPO}$$

MISION 3, el objetivo es llegar al final del nivel habiendo recogido las 10 piezas del mapa que completan el camino hacia la Fuente Mágica.

Por el camino, nos encontramos con obstáculos, el sistema de puntuación es el siguiente:

CADA MONEDA DA 1 PUNTO

CADA PUERTA DA 2 PUNTOS

CADA MURO DESTRUCTIBLE DA 5 PUNTOS

CADA OBSTACULO SALTADO DA 5 PUNTOS

CADA MURO MOVIL DA 10 PUNTOS

CADA PIEZA DEL MAPA DA 25 PUNTOS

El sistema de puntuación final se calcula de la siguiente forma:

$$\text{PUNTUACION} = (\text{MONEDAS} + \text{PUERTA} + \text{MUROS} + \text{SALTOS} + \text{PIEZA}) \times \text{VIDA} \times 100 / \text{TIEMPO}$$

4.11 CONTROL DE VERSIONES - GITHUB

Como se ha comentado anteriormente, se ha implementado un repositorio compartido en GitHub en el cual existe un control de cambios del proyecto.

Al tener el código compartido en la nube, nos daba versatilidad para trabajar en diferentes equipos, sin embargo, una fortaleza de este trabajo ha sido la comunicación entre los miembros del equipo sobre las partes del código en proceso de desarrollo para evitar realizar cambios que afecten el trabajo o codificación de quien este activo realizando modificaciones al proyecto.

5 GDD DE ALTO RENDIMIENTO

5.1 INTRODUCCIÓN

Ancient Galápagos se sitúa en las islas Galápagos ubicadas a 972 km de las costas del Ecuador en el océano Pacífico. Como punto de partida ubicamos a nuestro personaje en la isla Isabela aproximadamente hace 500 años, unos años antes de su descubrimiento. Existen un par de teorías sobre su descubrimiento y una de ellas es que estas islas fueron descubiertas por los incas mucho antes de su registro en el año 1535.

Ciertamente, de aspecto Lowpoly Cartoon ambientado en las 13 islas del archipiélago y los inicios de su colonización indígena por lo cual tomaremos la imagen de sus chozas y particularidades del pueblo indígena (shuar) para la apariencia de los pobladores, en un mundo rodeado por mar y con especies referentes a las encontradas en las islas, nos enfrentaremos a piratas con el fin de salvaguardarlas.

Ancient Galápagos nos pone en los zapatos de un aldeano que lucha contra enemigos piratas que tratan de saquear las especies de las islas, obteniendo habilidades mágicas y preparándose para su gran misión

5.2 ALCANCE DEL JUEGO

Este proyecto está ambientado en las 13 islas del archipiélago y los inicios de su colonización indígena, en un mundo rodeado por mar y con especies referentes a las encontradas en las islas, nos enfrentaremos a piratas con el fin de salvaguardarlas. Por tal razón, el videojuego consta de 3 niveles, correspondientes a 3 partes de la primera de las islas, isla Isabela, posteriormente el personaje se va moviendo a otras islas, (nuevas expansiones del juego o desarrollos dependiendo de la aceptación del juego en el mercado).

1er acto: Buscar información entre los aldeanos sobre qué puedes hacer en la isla, liberar especies, rescatar a George y alcanzar el estanque dorado para conseguir las piedras mágicas (arma arrojada de munición infinita).

2do acto: Con una nueva habilidad permite comunicar con el pequeño George y él ha informado que la gran tortuga necesita ayuda. Debes abrirte paso entre enemigos y obstáculos del escenario hasta llegar a la gran tortuga.

3er Acto: El gran pirata barba verde, se ha atrincherado en el santuario y lo está saqueando.
Debes detenerlo

5.3 ESTILO VISUAL DE ALTO NIVEL

El diseño artístico está enfocado en tipo cartoon con baja poligonización y colores muy vivos, así, se trabajará en el aspecto visual con tonos pasteles, donde se pretende crear una estética más agradable a lo tradicional. La Ilustración 4, de este trabajo pretende ser la principal referencia para la creación de este nuevo estilo visual que enmarcará radicales cambios en el proyecto.

El objetivo de hacer el juego asequible y vistoso para todo el público determina que el juego debe emplear en su imagen un rango de valores amplio, disponiendo de una gran variedad de gradientes, por consiguiente, la intencionalidad debe transmitir una invitación directa a explorar y descubrir el mundo desde cada escenario del videojuego.

Otro aspecto, es la combinación de color e iluminación en cada escenario de cual son elementos que trabajan continuamente para contrarrestar las limitaciones de la baja poligonización en los modelos 3D, donde, se irá transmitiendo al jugador con una alta inmersividad del juego, efectos de sombras y claroscuros, destacando contrastes de luz sobre el ambiente del juego.

Por otra parte, se podrá mantener el rango de valores en las ocasiones que se precise iluminar más la escena, por tal, se deberá abusar del color pastel tan característico de las películas de ciencia ficción porque generalmente se usa una temperatura de color fría y se destaca colores amarillos (representando energía, vitalidad acción), marrones y verdes con los cual identificamos como el abandono del control del ser humano sobre el mundo permitiendo que la naturaleza avance y evolucione.

5.4 INTERFAZ

❖ Control

El juego utilizará en su versión de computador (versión a prototipar), teclado y ratón para la interacción del usuario con el personaje principal, detallada nuevamente en la Tabla 11:

Tabla 11. Control del videojuego.

Tecla del computador	Acción a realizar
W, Flecha Arriba	Movimiento adelante
S, Flecha Abajo	Movimiento atrás
A, Flecha Izquierda	Movimiento a la izquierda / giro izquierdo
D, Flecha Derecha	Movimiento a la derecha / giro derecho
Barra Espaciadora	Saltar
Q	Ver inventario
Botón Derecho del Ratón	Agarrar / Soltar / interactuar con objeto
Shift	Caminar despacio en runner / Correr
E	Usar Objetos desde inventario

Fuente: Elaboración propia.

❖ Cámara

La cámara determina por el uso de la llamada "Third-Person View", esta vista tiene como característica principal que el personaje que se controla se ve de cuerpo entero y generalmente de espaldas, por tal razón, es ideal para juegos en los que la acción se basa más en la búsqueda de objetos o indicios para descubrir la trama del juego, generando mayor sensación de inmersión.

Una de las ventajas que presenta esta vista es que el jugador puede ver directamente los alrededores del personaje que controla sin tener que voltear la vista y el control de la cámara se realiza con el movimiento del ratón usando los ejes X (Yaw) y Y (Pitch). Siendo que, el eje Y queda limitado por los ángulos +30 grados y -10 grados, de forma que la cámara no puede quedar visualizando ángulos de difícil manejo para el jugador (Computer Hope, 2022).

El personaje principal de este proyecto cuenta con dos cámaras: La primera (Cámara por defecto) sigue al personaje por la parte de atrás a una distancia de dos metros, y la cámara secundaria se encuentra medio metro delante del personaje a la altura del pecho y con la tecla

"Z" se puede realizar un switch entre estas dos cámaras, pasando de primera persona a tercera persona.

❖ HUD

La interfaz de usuario y el principal panel de comunicación con el jugador permite que este obtenga feedback constante sobre su situación en el juego, así, se considera el desarrollo de un HUD tradicional superpuesto a la pantalla de juego.

Como se puede observar en la Figura 42, se ha implementado (por falta de recursos y tiempo) un HUD de depuración para visualizar el correcto funcionamiento. Este HUD implementado ocupa la parte superior de la pantalla e informando al jugador de:

- ❖ **Vidas que le quedan:** Veces que va a poder regenerarse en el nivel una vez muerto
- ❖ **Vida actual:** Una barra de color rojo que permita visualmente entender la situación actual de la vida del jugador.
- ❖ **Enemigos eliminados:** Un contador que indica el número de enemigos eliminado
- ❖ **Puntuación:** Un contador acumulado de los puntos actuales del jugador
- ❖ **Contadores individuales:** Que identifican la cantidad de especies rescatadas
- ❖ **Contador de munición (piedras recogidas):** Un contador de referencia para el jugador para que conozca de primera mano cuánta munición tiene disponible.
- ❖ **Contador de Frutas recogidas:** Identifica las piezas de fruta de recogidas y las que faltan para poder recuperar una vida perdida o recuperar parcialmente la actual vida en la barra de vida.
- ❖ **Contador de piezas del mapa recolectadas:** Contador que identifica cuantas piezas se han recogido y cuantas quedan para poder completar el mapa correspondiente.

Figura 42. HUD Debug para el videojuego.



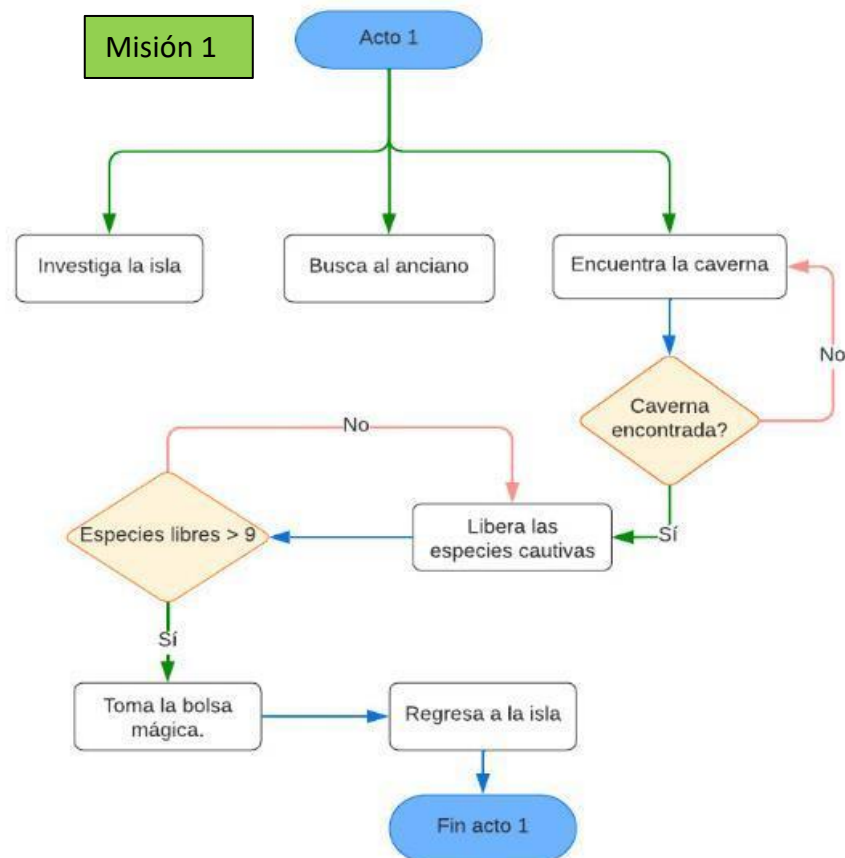
Fuente: Elaboración propia.

5.5 FLUJO GENERAL

El jugador quien posee una vista en tercera persona, puede guiar a "Inti" en las misiones que debe completar para terminar el nivel de turno, el cual debe enfrentar a enemigos en combates ya sea cuerpo a cuerpo o a distancia, también debe ir resolviendo puzzles, rescatando especies, recorriendo distancias largas o escabulléndose para evitar ser detectado y así completar a satisfacción cada nivel.

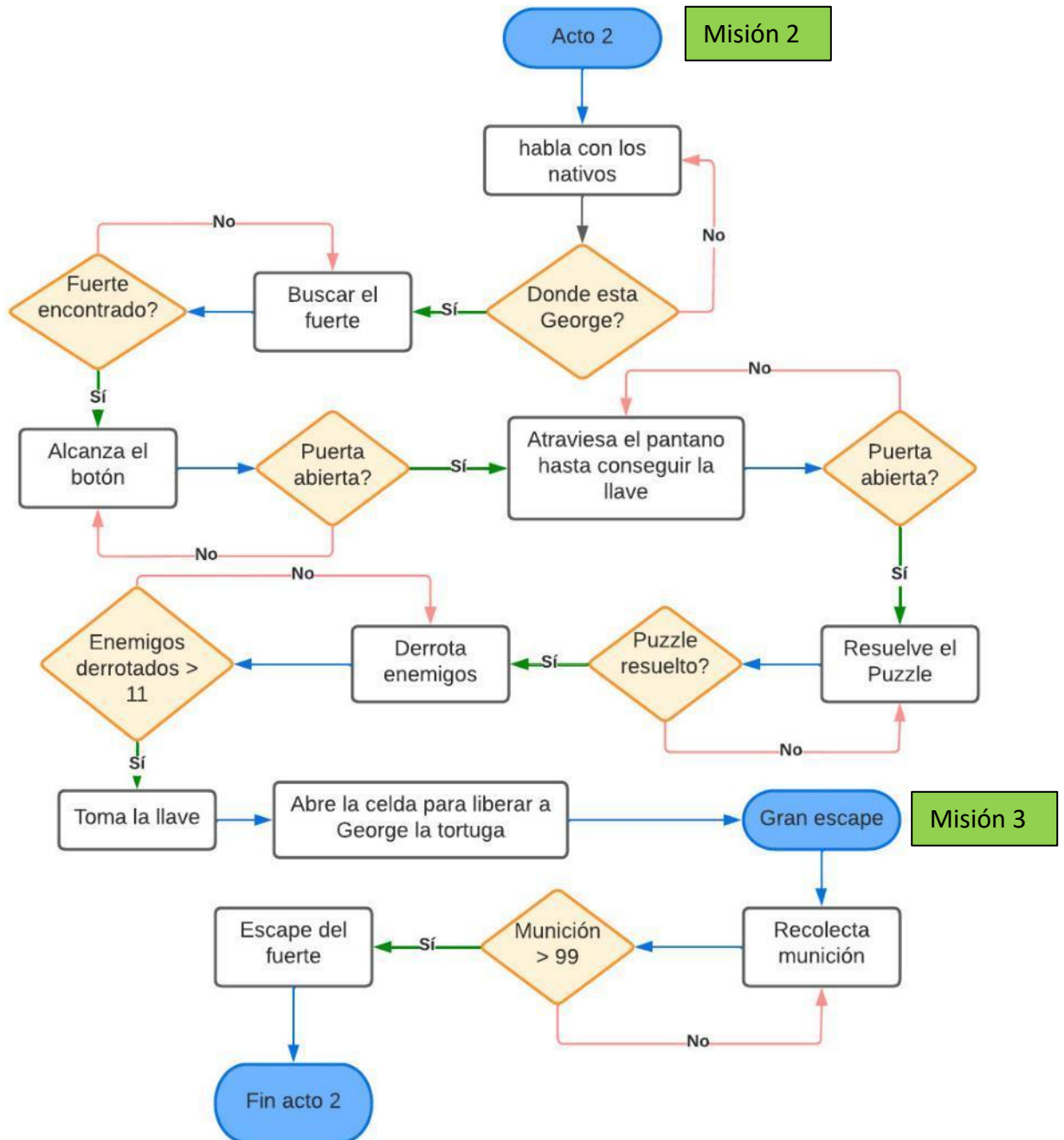
El nivel 1 queda dividido en 3 misiones principales: rescatar especies, mostrado en la Figura 43, liberar a George y alcanzar el Estanque Mágico, en la Figura 44. Cada uno de esos niveles, tal y como se quedará detallado en el apartado 5.6 tienen diferentes fases. En la Figura 45, se muestra el flujo del nivel 3 que no está implementado en el demo jugable por cuestiones de faltas de recursos de equipo, tiempo y dinero.

Figura 43. Flujo del Nivel 1 – Misión 1.



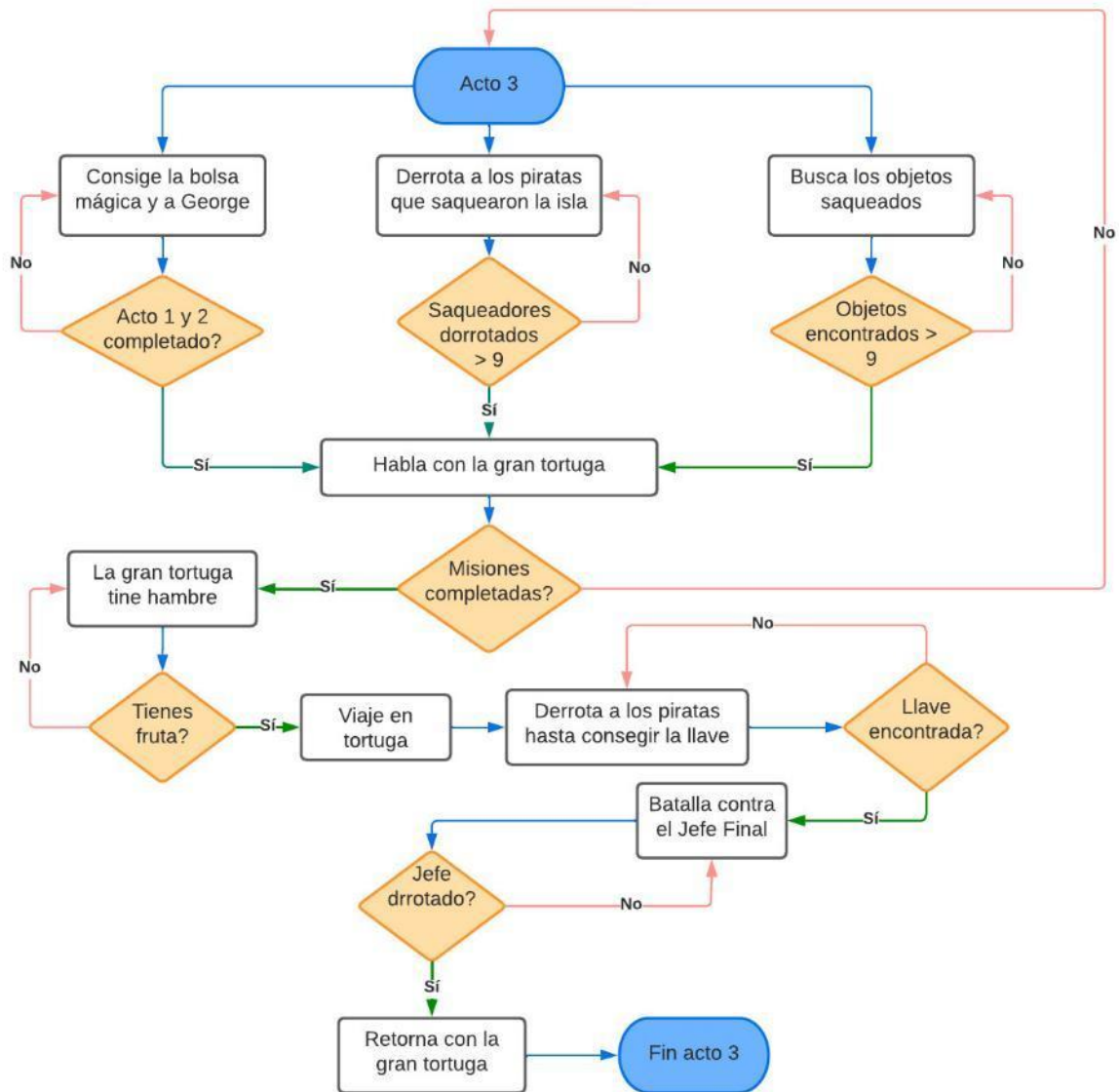
Fuente: Elaboración propia.

Figura 44. Flujo del Nivel 1 – Misión 2 y 3.



Fuente: Elaboración propia.

Figura 45. Flujo de Nivel 2 y 3 (No implementado).



Fuente: Elaboración propia.

5.6 DISEÑO DE NIVELES

❖ Lore – Intro Inicial

Mensaje por pantalla (George): *Cálmate y escucha... sigue avanzando y libera a mi amigo! ¡no hay tiempo!*

Mensaje por pantalla (Inti): *Los veo! tiene guardias y podría ser peligroso si nos ven, sería mejor escabullirse sin ser visto hasta liberarlo. ¡¡Luego hacerlos que paguen!!*

Mensaje por pantalla (George): ¡Excelente Inti! ahora estoy seguro que eres el elegido, es hora que conozcas a uno de los espíritus de esta isla... ¡Busca a la gran tortuga!

Mensaje por pantalla (Inti): No he visto una gran tortuga!

Mensaje por pantalla (George): Seguro que sí, vosotros le llaman: ¡la gran roca!

Mensaje por pantalla (George): Dile que vas de mi parte, dile que el guardián ha despertado"

Siendo este el inicio del juego comienza siendo el primer momento en el que el jugador interactuar con el entorno del videojuego.

❖ Valor de refuerzo de la misión

En todo momento e independientemente de en qué misión se encuentre el jugador, este puede preguntar a los aldeanos sobre qué puede hacer, además existen Letreros que también se bastan como refuerzo permitiendo indicar al jugador la dirección que debe tomar, en consecuencia, en cada misión al encontrar letreros darán pistas sobre que se debe realizar en cada desafío que se presenta en el camino.

5.6.1 Nivel 1

El diseño del nivel 1 se describe de la siguiente manera, siendo el nivel que se ha implementado como parte principal en el prototipo, tendrá el mayor énfasis de realismo para que pueda ser fácilmente replicable a los siguientes niveles.

El nivel 1 comienza con un video de intro donde aparecen las especies, una vista a vuelo de pájaro sobre la primera isla y nuestro personaje Inti. En esa escena se pueden observar las diferentes props y NPCs definidos anteriormente (especies en extinción, cajas de rescate, chozas, nativos y piratas).

5.6.1.1 Inicio del juego, Nivel 1, pantalla principal, acción 1

El escenario base del Nivel 1 es la Isla Isabella, es la primera isla del archipiélago y es el entorno en el cual se desarrolla la acción a través de 3 diferentes misiones.

También, en nuestro caso, es el centro del gameplay, y, por lo tanto, la hemos descrito con más detalle.

Este escenario es la pantalla principal que da acceso a las misiones y es el hilo conductor de la narrativa de la experiencia de usuario.

Inti se encuentra en Isabella y debe avanzar sobre ella para avanzar en el juego.

Resumen: Conversar con los aldeanos hasta conseguir pistas sobre que está sucediendo, las pistas conseguidas te llevaran a las misiones, una vez completada las misiones podrás desbloquear el diálogo con La Gran tortuga que te dará el acceso a la siguiente isla.

Enemigos: No están disponibles. Se muestra más detalle en la Figura 46:

Figura 46. Mapa de la Isla Isabella.



Fuente: Elaboración propia.

Interacciones:

- Puntos de información principal: A modo de tutorial le presenta un resumen de las acciones disponibles y el objetivo principal.
- Puntos de información: Presenta breves consejos sobre los puntos o personajes más relevantes que hay por la zona y los requisitos para acceder a una misión en caso de existir una misión disponible.
- NPC: Patrullan una o varias zonas al azar en un rango determinado, cada cierto tiempo se detienen, tienen al menos tres tipos de diálogos aleatorios, los diálogos suelen ser consejos para completar alguna misión o cosas irrelevantes, los diálogos largos solo están disponibles si están en modo quieto.
- Misión 1: Liberar las especies. Disponible desde el inicio del juego te lleva a la misión rescate de las especies.
- Misión 2: Rescate de George la tortuga. Disponible desde el inicio del juego, solo se puede acceder si se completa la misión 1.
- La gran tortuga: Solo disponible si se ha completado la misión 1 y 2, se desbloquea el dialogo entre el jugador y la tortuga gigante.

Condición de Victoria: Completa las misiones.

Mecánicas: Caminar, Saltar y Correr.

Condición de derrota: No disponible.

5.6.1.2 Nivel 1, misión 1, liberar las especies

Inti, tras hablar con un aldeano descubre que se están secuestrando especies y debe de rescatarlas, esto significa que debe dirigirse a una zona de teletransporte (zona marcada como **Misión 1 Liberar las especies** en la Figura 46).

Esta misión consta de 3 fases que requieren de diseño y se detallan a continuación:

Título: Nivel 1 - Misión 1 - fase 1 – MODO ENTRENAMIENTO - RESCATAR LA ESPECIE LEON MARINO.

Resumen: El jugador debe aprender la mecánica de agarrar y soltar objetos. El objetivo principal es agarrar una caja y llevarla hasta la zona de rescate. El jugador aprende a usar las dinámicas agarrar y soltar cajas mediante el botón DERECHO del ratón. Ver Figura 47:

Enemigos: No existen generadores de piratas activos en esta misión, tampoco existen piratas itinerantes ni patrullando.

Cajas: Nada más empezar la caja está posicionada y preparada para que el jugador interactúe. No existen generadores de cajas,

Interacciones:

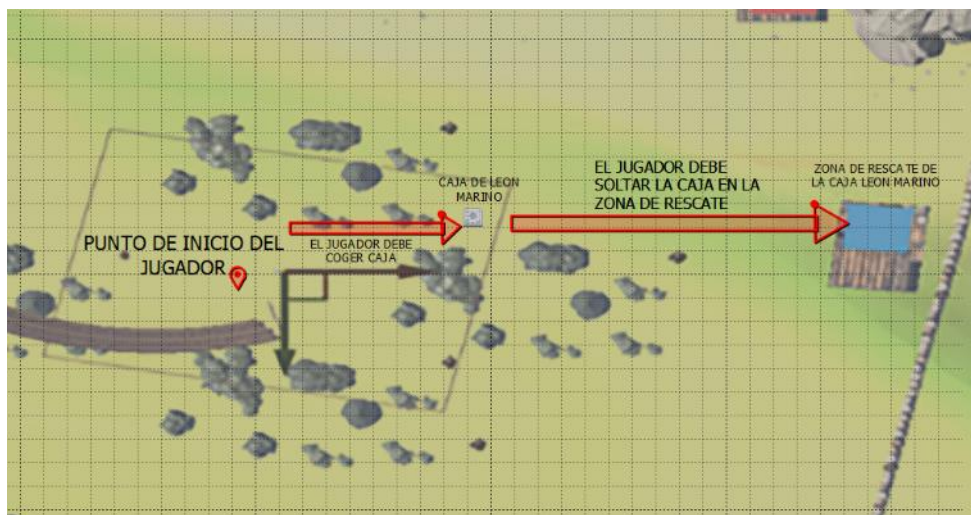
- Usar el botón DERECHO del ratón para coger una CAJA
- Desplazarse con las FLECHAS con la caja hasta la zona de RESCATE
- Usar el botón DERECHO del ratón para soltar la caja en la zona adecuada.

Condición de victoria: Dejar correctamente la caja en la zona de rescate

Mecánicas: Moverse, Correr, Saltar, Agarrar y Soltar.

Condición de derrota: Si el jugador no desplaza la caja a la zona de rescate, no hay éxito en la misión.

Figura 47. Mapa de la misión 1 - fase 1 del Nivel 1.



Fuente: Elaboración propia.

Título: Nivel1 - Misión 1 - fase 2 – MODO ENTRENAMIENTO - ELIMINAR UN ENEMIGO Y RESCATE DE ESPECIE

Resumen: Continuando con la curva de aprendizaje del jugador, en esta fase, el jugador aprender la mecánica de lanzar cajas para eliminar enemigos. Para ello, debe enfrentarse a un único pirata lanzándole una caja como arma. Después de destruir al enemigo, el jugador debe volver a coger la caja y dirigirse a la zona de rescate. Ver Figura 48:

Enemigos: Existe un único pirata posicionado previamente y en modo patrulla. No existen generadores de piratas activos en esta misión.

Cajas: Nada más empezar la caja está posicionada y preparada para que el jugador interactúe. No existen generadores de cajas.

Interacciones:

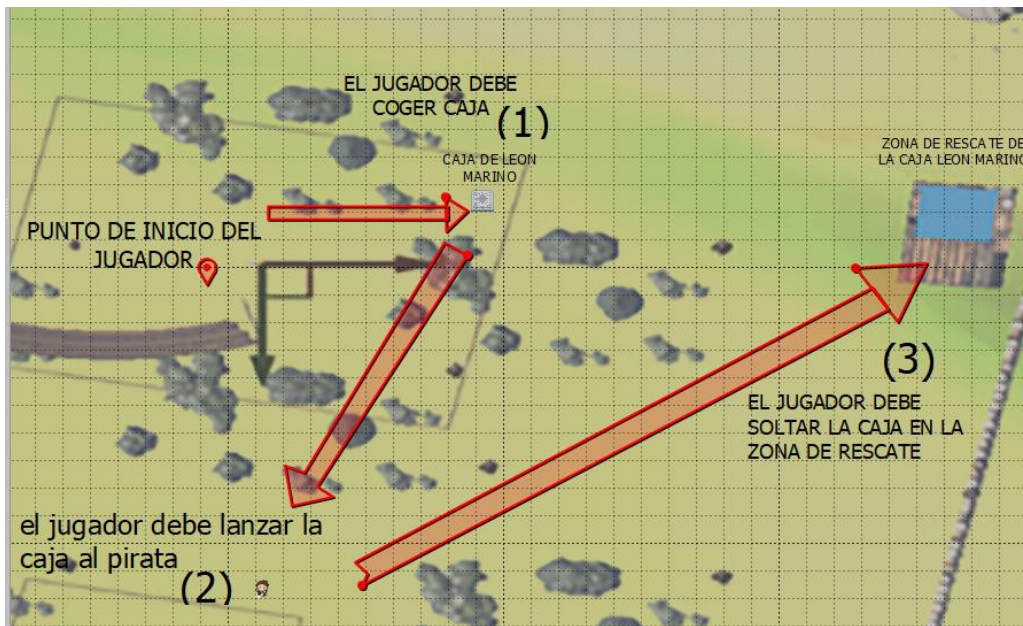
- Usar el botón DERECHO del ratón para agarrar una CAJA
- Desplazarse con las FLECHAS con la caja hasta la posición del pirata
- Usar el botón IZQUIERDO del ratón para LANZAR la caja contra el enemigo.
- Una vez destruido, usar el botón DERECHO para volver a agarrar la caja
- Usar las FLECHAS para desplazarse hasta la zona de RESCATE
- Usar el botón DERECHO del ratón para soltar la caja en la zona adecuada.

Condición de victoria: Eliminar primero al pirata y posteriormente dirigirse a la zona de rescate y dejar correctamente la caja en la misma.

Mecánicas: Moverse, Correr, Saltar, Coger, Lanzar y Soltar.

Condición de derrota: Si el jugador no elimina al pirata, o el pirata elimina la vida del jugador o si el jugador NO desplaza la caja a la zona de rescate tras eliminar al pirata, se considerará que no hay éxito en la misión.

Figura 48. Mapa de la Misión 1 – fase 2 del Nivel 1.



Fuente: Elaboración propia.

Título: Nivel 1 - Misión 1 - fase 3 – MODO RETO – RESCATE DE 5 CAJAS DE CADA ESPECIE (IGUANAS, LEONES MARINOS Y TORTUGAS)

Resumen: En esta misión, en principio el jugador ya conoce las dinámicas necesarias para poder jugar y es el momento de poner a prueba su habilidad jugando. Existen 3 zonas de rescate con diferente nivel de dificultad: LEONES MARINOS (FACIL), IGUANAS (MEDIO) Y TORTUGAS (DIFICIL). Ver Figura 49:

Enemigos: Existen 3 zonas de Spawn de enemigos con una IA muy agresiva, se dirigen directamente hacia el jugador con el objetivo de restarle vidas. Para que la curva de dificultad no aumente proporcionalmente, el número de enemigos en la arena será igual al número de cajas que se han rescatado + 1. Digamos que, cuando todavía no se hayan rescatado ninguna caja, solo hay un enemigo, cuando llevemos 14 cajas y quede una, tendremos 15 enemigos persiguiéndonos. De la misma forma, la velocidad de los enemigos irá en aumento conforme vayamos salvando especies.

Cajas: En el centro del nivel se encuentra localizado un Spawn de cajas que cada 10 segundos suministrará una caja nueva en una zona aleatoria marcada. Como las cajas no pueden destruirse en 150 segundos, estarán todas las cajas sobre el escenario (5 de cada especie).

Interacciones:

- Usar el botón DERECHO del ratón para agarrar una CAJA.
- Desplazarse con las FLECHAS con la caja hasta la posición del pirata.
- Usar el botón IZQUIERDO del ratón para LANZAR la caja contra el enemigo.
- Usar el botón DERECHO para volver a agarrar la caja.
- Usar las FLECHAS para desplazarse hasta la zona de RESCATE.
- Usar el botón DERECHO del ratón para soltar la caja en la zona adecuada.

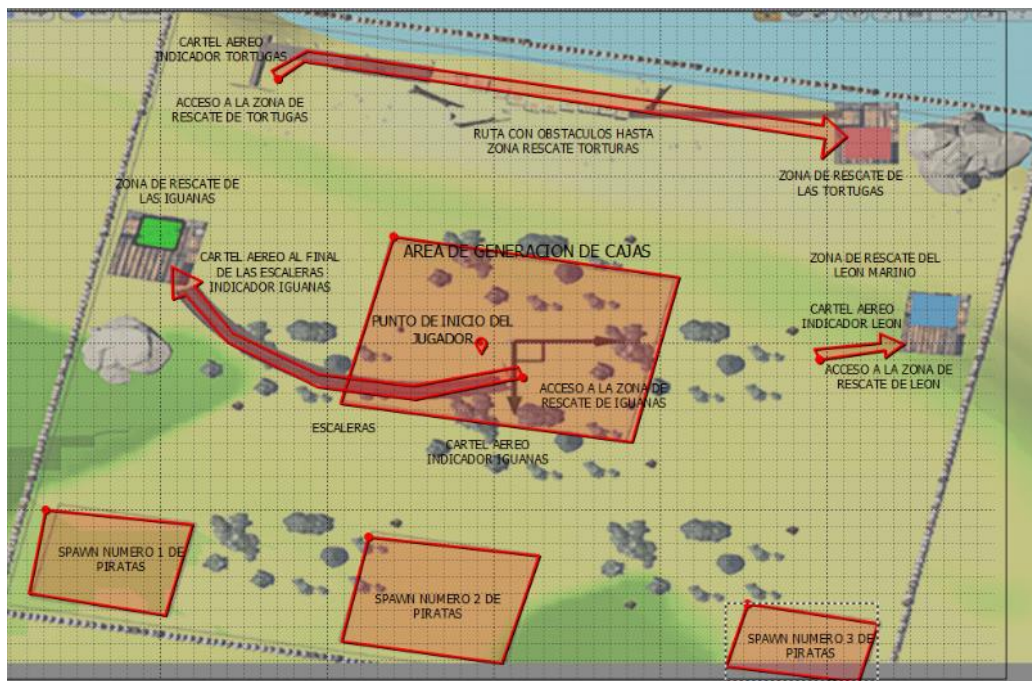
Condición de victoria: Rescatar 15 especies, considerando 5 de cada tipo (5 IGUANAS, 5 LEONES MARINOS Y 5 TORTUGAS).

Mecánicas: Moverse, Correr, Saltar, Agarrar, Lanzar y Soltar.

Condición de derrota: Si un pirata enemigo elimina la vida del jugador o si el jugador NO desplaza 15 cajas a las respectivas zonas de rescate se considerará que no hay éxito en la misión.

Recompensa por logro: a Up. El jugador consigue un sistema de disparo. Una mochila llena de piedras con la cual poder atacar enemigos a distancia sin necesidad de las cajas.

Figura 49. Mapa de la Misión 1 – fase 3 del Nivel 1.



Fuente: Elaboración propia.

Otra información para tener en cuenta en el diseño de la fase: Existen 3 zonas de rescate en el nivel, una para cada especie, allí hay unos nativos esperando, encargados de ayudar a las especies capturadas, cuidarlas y devolverlas al ambiente natural cuando sea necesario, este es el motivo por el cual, las especies rescatadas por ataque a piratas no se consideran rescatadas.

La primera zona de rescate (rescate de iguanas) muy sencilla, consistiendo en una rampa en el ala izquierda del nivel. La idea es que el jugador comience a tener manejo de los controles para las mecánicas de coger, lanzar y soltar cajas.

La segunda zona es algo más complicada, permite rescatar leones marinos, se accede a través de un laberinto, el jugador debe practicar la destreza con los controles, la dificultad principal radica en la entrada y salida del laberinto ya que en las mismas pueden existir piratas en la ruta.

La tercera ruta consiste en practicar las destrezas de las mecánicas básicas combinadas con el salto, donde, el jugador deberá empujar bloques y crear su propia ruta para poder acceder al tercer punto de rescate. Es decir, esta zona rescata tortugas y tras rescatar a la última de las tortugas, los nativos le dicen que allí no está el pequeño George, una tortuga muy importante. Así que debe rescatarla, los nativos le dicen que creen que está capturada en una zona especial de la isla, protegida de grandes enigmas, así, nuestro personaje se debe dirigir allí.

5.6.1.3 Transición, Nivel 1, pantalla principal, acción 2

Resumen: Una vez rescatadas las especies y conseguido el powerup MOCHILA, el jugador vuelve a la pantalla principal, donde se encuentra con otro NPC que le indica que no ha logrado rescatar a GEORGE, la tortuga más importante de la isla, y que, por lo que tiene entendido, se encuentra custodiada en una fortaleza en el interior de la isla.

Inti se debe dirigir a la fortaleza y entrar.

La localización exacta del NPC y la fortaleza se encuentra indicada en el apartado descrito anteriormente y titulado "Inicio del juego, Nivel 1, pantalla principal, acción 1".

5.6.1.4 Nivel 1, misión 2, rescate George

Título: Nivel 1 - Misión 2 - fase 0 – INTRODUCCION DE NUEVAS DINAMICAS

Resumen: Tras pasar los umbrales de la fortaleza, el jugador se encuentra con una primera sala muy simple donde va a aprender las principales dinámicas a usar en esta misión 2. Por lo tanto, el jugador se encuentra ante una habitación donde se pondrá en práctica las mecánicas necesarias para poder completar la misión. Ver Figura 50:

Enemigos: Este nivel no dispone de enemigos, se encontrará con elementos estáticos, a los cuales tendrá que disparar.

Interacciones:

- Salto, empleando la tecla ESPACIO.
- Desplazamiento, empleando las teclas asignadas.
- Llave: agarrar el objeto llave para abrir la puerta usando el BOTÓN DERECHO del ratón.
- Puerta: Al acercarse a la puerta y si tiene la llave esta se abre y permite el paso al siguiente desafío.
- Dianas de tiro: Se destruyen con las piedras que se lanzan.
- Lanzar piedras: Usando el botón IZQUIERDO del ratón, INTI podrá lanzar piedras de la recién recogida mochila.
- Cajas que se pueden empujar para usarlas como plataformas. La interacción se produce moviéndonos en la dirección donde se encuentran las cajas, las desplazamos con nuestro movimiento.
- Caja: Se puede empujar para desbloquear el paso a la puerta.
- Salida del nivel: abriendo la puerta.
- Punto de información: Le cuenta al jugador el objetivo del nivel y los peligros de este.

Condición de Victoria: Destruir los objetos marcados, recoger la llave. Desplazar elementos para crear el camino de salida hacia la puerta y Abrir la puerta.

Condición de Derrota: Inti no avanza en el nivel si no ejecuta alguna de las condiciones de Victoria.

Mecánicas: Caminar, Correr, Salto, Empujar, Agarrar, Usar y Lanzar.

Figura 50. Mapa de la Misión 2 – fase 0 del Nivel 1.



Fuente: Elaboración propia.

Título: Nivel 1 - Misión 2 - fase 1 – SALTAR Y EMPUJAR

Resumen: El objetivo principal es alcanzar la puerta para pasar a la siguiente habitación. Para ello, el jugador deberá familiarizarse con el salto y empujar las cajas de manera que le permitan llegar a un interruptor que se encuentra en lo alto de una plataforma. Ver Figura 51:

Enemigos: Este nivel no dispone de enemigos.

Interacciones:

- Salto, empleando la tecla ESPACIO.
- Desplazamiento, empleando las teclas asignadas.

- Cajas que se pueden empujar para usarlas como plataformas. La interacción se produce moviéndonos en la dirección donde se encuentran las cajas, las desplazamos con nuestro movimiento.
- Botón que abre la puerta. La interacción se produce posicionándonos en la posición donde se encuentra el botón.
- Puerta que permite el acceso al siguiente nivel. La interacción se produce si se pulsa el botón.
- Punto de información: Le cuenta al jugador el objetivo del nivel y sus peligros.

Condición de Victoria: Abrir la puerta pulsando el botón que esta al final de las plataformas.

Mecánicas: Caminar, Correr, Salto, Empujar.

Condición de derrota: Si el jugador no logra alcanzar el interruptor y como consecuencia no pasa la puerta hacia la siguiente habitación.

Figura 51. Mapa de la Misión 2 – fase 1 del Nivel 1.



Fuente: Elaboración propia.

Título: Nivel 1 - Misión 2 - fase 2 – ENCUENTRA LA LLAVE

Resumen: Segundo nivel, el jugador ya con más destreza saltando, debe atravesar el pantano, entre las rocas y plataformas móviles, hasta llegar a la llave que le permitirá pasar al siguiente nivel. Si en el camino el jugador cae, se posicionará en el inicio de la fase 2. Ver Figura 52:

Enemigos: Este nivel no dispone de enemigos.

Interacciones:

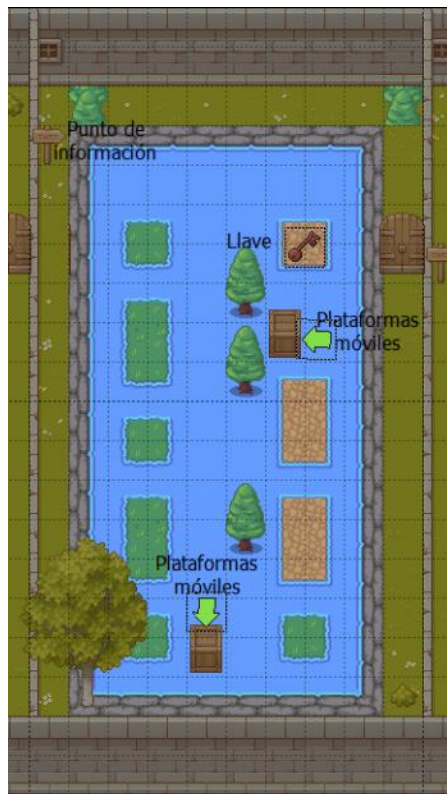
- Salto, empleando la tecla ESPACIO.
- Desplazamiento, empleando las teclas asignadas.
- Pantano si te caes en el aparecerás al inicio de la fase.
- Plataformas móviles: Se desplazan de un punto a otro y de retorno de manera indefinida.
- Agarrar la llave: Usa el botón DERECHO del ratón para agarrar la llave.
- Abrir puerta: Usa el botón DERECHO del ratón para interactuar con la puerta y abrir.
- Punto de información: Le cuenta al jugador el objetivo del nivel y sus peligros.

Condición de Victoria: Obtener la llave para abrir la puerta.

Mecánicas: Caminar, Correr, Salto, Agarrar y Usar

Condición de derrota: Si el jugador no logra acceder a la puerta de acceso a la siguiente habitación.

Figura 52. Mapa de la Misión 2 – fase 2 del Nivel 1.



Fuente: Elaboración propia.

Título: Nivel 1 - Misión 2 - fase 3 – EL ENIGMA

Resumen: Tercer nivel, Resuelve el acertijo. En este nivel el usuario se enfrenta por primera vez con un tipo de PUZZLE que debe resolver, intentar interpretar las pistas que se le dan para activar en el orden correcto los botones. Ver Figura 53:

Enemigos: Este nivel no dispone de enemigos.

Interacciones:

- Pista del acertijo: El muro contiene una pista de la secuencia del acertijo.
- Botones acertijo: Dispone de 4 botones con sonido de acierto y de error, cambio de colores entre verde para identificar acierto y rojo para error, se reinician con el error mientras que en el acierto se mantienen su color.
- Abrir puerta: Al acercarse a la puerta y si se ha resuelto el acertijo esta se abre.
- Punto de información: Le cuenta al jugador el objetivo del nivel.

Condición de Victoria: Resolver el acertijo para abrir la puerta.

Mecánicas: Caminar.

Condición de derrota: No disponible, los intentos son indefinidos.

Figura 53. Mapa de la Misión 2 – fase 3 del Nivel 1.



Fuente: Elaboración propia.

Título: Nivel 1 - Misión 2 - fase 4 – EL RESCATE DE GEORGE

Resumen: Cuarto nivel, Combate. En este nivel el usuario se enfrenta por primera vez con el combate sin cajas. El objetivo principal es rescatar a George, pero para ello debe enfrentarse a unas oleadas de los enemigos hasta conseguir la llave que abre la celda que mantiene cautivo a George la tortuga y poder escapar con él. Ver Figura 54:

Enemigos: 4 oleadas de 3 enemigos.

Interacciones:

- Salto, empleando la tecla ESPACIO.
- Desplazamiento, empleando las teclas asignadas.
- Enemigos: Los enemigos patrullan la zona, si el jugador está en rango de visión del enemigo o lo ataca este inmediatamente corre hacia el para atacarlo, duplicando su velocidad por un breve lapso de tiempo.
- Spawn enemigos: Son puntos en el mapa donde salen los enemigos de manera aleatoria.
- Llave: Solo aparece cuando se han derrotado todas las oleadas de enemigos y sirve para abrir la puerta.
- Puerta: Al acercarse a la puerta y si tiene la llave esta se abre.
- Salida del nivel: Solo si ha rescatado a George la Tortuga podrá salir del nivel hacia el siguiente nivel "El Gran escape".
- Punto de información: Le cuenta al jugador el objetivo del nivel y los peligros de este.

Condición de Victoria: Derrota las oleadas de enemigos hasta obtener la llave y rescatar a George la tortuga.

Mecánicas: Caminar, Saltar, Atacar y Correr.

Condición de derrota: Si el jugador pierde su vida aparece al inicio del cuarto nivel y se reinician las oleadas.

Al finalizar el nivel, Inti se ha quedado sin piedras en la mochila, George le dice que hay una fuente con unas piedras mágicas que le darían munición infinita de piedras.

Recompensa por la victoria: Primer trozo del mapa que le permite comenzar la carrera hacia la fuente.

Figura 54. Mapa de la Misión 2 – fase 4 del Nivel 1.



Fuente: Elaboración propia.

5.6.1.5 Nivel 1, misión 3, Carrera por alcanzar la Fuente Mágica

Título: Nivel 1 - Misión 3 - fase 1 – ALCANZA LA FUENTE

Resumen: Empezamos una carrera hacia la fuente mágica, el objetivo es correr sin parar hasta conseguir un número determinado de piezas del mapa que nos lleva hacia la misma, si no las recogemos, parecerá que nos hemos perdido en el camino.

En el camino hacia la fuente deberemos evitar múltiples obstáculos como paredes, puertas, enemigos, huecos. Algunos obstáculos se pueden destruir y otros son indestructibles, por lo que el jugador debe esquivar para evitar perder vida, cada vez que pierde vida, se reinicia desde el comienzo del nivel perdiendo las partes del mapa conseguidas.

Enemigos: Los enemigos harán las veces de torreta y serán fijos.

Recolecta de Piezas de Mapa: En el camino nos encontraremos piezas de mapa que deberemos recoger para poder completar la misión. Estas piezas son colocadas de forma aleatoria.

Recolecta de Fruta: En el camino podremos recolectar piezas de fruta que podremos usar para recuperar vida parcial (uso instantáneo) o acumular para recuperar vidas. Estas piezas son colocadas de forma aleatoria.

Recolecta de Piedras: También de forma aleatoria, nos encontraremos con piedras que el jugador puede recoger, acumular y/o lanzar para destruir obstáculos.

Objetos destructibles: Existirán objetos destructibles que podrán ser eliminados al momento de colisionar con una piedra lanzada. Los objetos pueden obstaculizar parcial o totalmente la ruta de acceso al camino.

Interacciones:

- Salto, empleando la tecla ESPACIO.
- Desplazamiento, empleando las teclas asignadas, usamos las teclas A y D para desplazar lateralmente al personaje a IZQUIERDA y DERECHA RESPECTIVAMENTE.
- Colisión con Objetos
- Destrucción de objetos
- Recolecta de Frutas, la recolección se produce por simple colisión del objeto con el player.
- Recolecta de Piedras, la recolección se produce por simple colisión del objeto con el player.
- Recolecta de piezas de mapa, la recolección se produce por simple colisión del objeto con el player.
- Lanzar piedras, si en el contador de PIEDRAS existe una cantidad superior a 0, el jugador puede lanzar piedras usando el BOTON IZQUIERDO.
- Usar FRUTAS para recuperar vida o obtener vidas extra. Cuando el jugador pulse la letra E, se consumirán todas las FRUTAS acumuladas. Primero se rellenará la barra de vida. Si todavía hay frutas tras tener la barra de vida a tope, si el jugador dispone de 10 piezas de fruta adicionales, podrá generar una vida extra.

Condición de victoria: Lograr recoger todas las piezas del mapa necesarias para alcanzar la fuente mágica.

Mecánicas: Correr, Saltar, Agarrar, Lanzar, Usar, Desplazamiento lateral a izquierda, Desplazamiento lateral a Derecha.

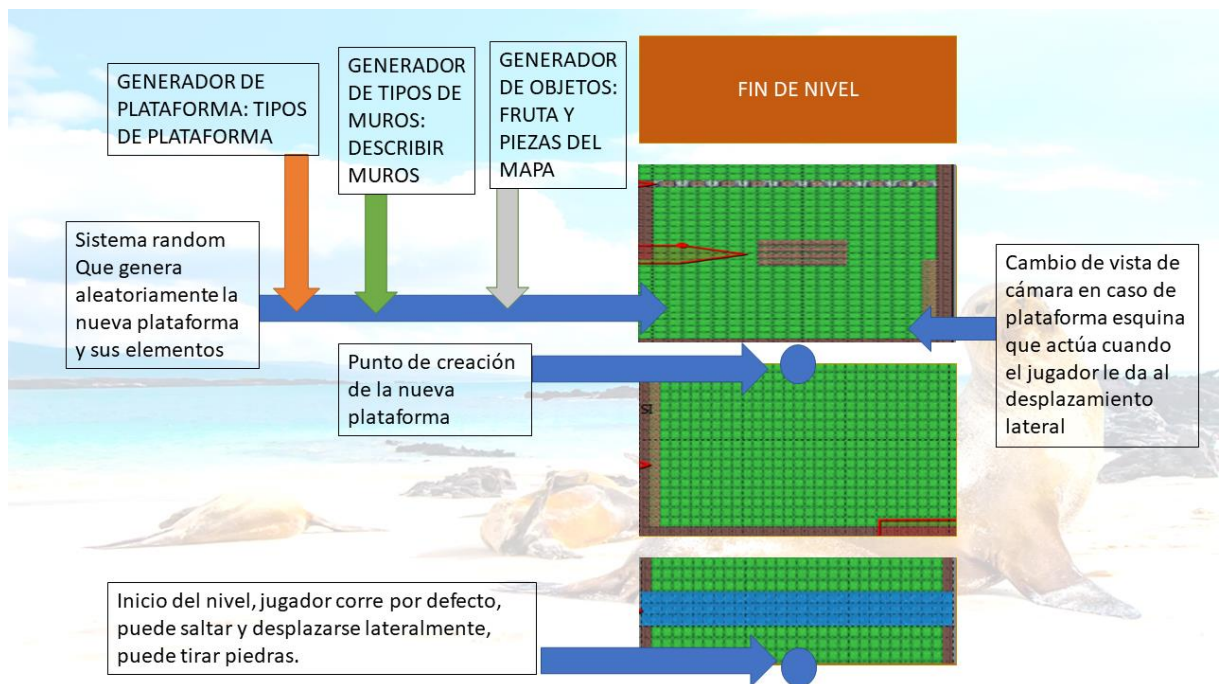
Perdida de vida:

Condición de derrota: Si el jugador pierde todas las vidas, el jugador caerá "muerto" en medio del juego.

Generación del nivel: El nivel se genera dinámicamente como los obstáculos y las monedas, existen plataformas inclinadas con elevación y depresión para hacer más dinámico el nivel.

Debido a la generación dinámica de este nivel, se realizó un resumen de su creación más específico, el cual debe evitar los choques contra las paredes, caerse en los huecos, chocarse con los obstáculos y todo lo anteriormente mencionado, se ubica el resumen de la implementación en la Figura 55:

Figura 55. Resumen de la Misión 3.



Fuente: Elaboración propia.

5.6.2 Nivel 2

Mecánicas y experiencia de usuario que aportan significativamente a este nivel del proyecto.

Mecánicas

- ❖ (añadidas a las anteriores)
- ❖ Recolección de ítems.
- ❖ Viaje por el mar a lomos de la tortuga.

Experiencia de usuario

- ❖ Derrota las hordas de enemigos hasta limpiar la aldea.
- ❖ Mueve los tótems a su sitio correspondiente (Los piratas con su sed de destrucción han movido los tótems que servían de guía para los caminantes).
- ❖ Recupera los objetos robados por los piratas y devuélvelos a los tótems.
- ❖ Habla con la gran tortuga (Ha pasado mucho tiempo y esta solo pude pensar en comida, te ha pedido una fruta cualquiera solo llévasela).
- ❖ La gran tortuga quiere probar tu valía y te llevará sobre su caparazón hasta una zona inaccesible de la isla, donde un temible pirata está causando estragos "Boss".

5.6.3 Nivel 3

Mecánicas y experiencia de usuario que aportan significativamente a este nivel del proyecto.

Mecánicas:

- ❖ Todas las anteriores, aunque principalmente disparar, moverse

Experiencia de usuario:

- ❖ Derrota a los enemigos hasta conseguir la llave del fuerte.
- ❖ Derrota al jefe final de la isla.
- ❖ Retorna con la gran tortuga.
- ❖ Emprende el viaje a la siguiente isla. Continuará

5.7 EL JUGADOR

"Inti" es el héroe del videojuego, con apenas 12 años de edad es seleccionado por la Pachamama (idioma nativo quechua, Pacha: mundo y mamá: Madre, La diosa de la tierra), para defender las especies de su pueblo, siendo un huérfano a la temprana edad de 5 años es criado por un anciano de la isla y que se gana la vida haciendo favores a los aldeanos del lugar. Siendo descendiente de padres shuar (idioma nativo de la isla) sueña con ser un guerrero y defender la isla de todos los que pretendan dañarla, así, pasa su tiempo libre visitando las especies que habitan en las islas jugando con ellas, tiene como amigo una tortuga que la llamo George y visita a diario, se dice que esta tortuga es una de las especies míticas del lugar.

5.7.1 Comportamiento

Las mecánicas básicas del personaje principal "Inti", se detallan a continuación:

Movimientos:

- ❖ **Caminar:** "Inti" puede moverse hacia delante, atrás, derecha e izquierda sin inercia, si el jugador cae de plataformas o de sitios altos se puede direccionar la caída tomando en cuenta la fricción por la caída.
- ❖ **Correr:** Duplica su velocidad de movimiento para desplazarse en las diferentes direcciones excepto hacia arriba, ahí solo se puede realizar la acción cuando el personaje está en el suelo.
- ❖ **Saltar:** Se eleva el doble de su tamaño por 2 segundos, también puede realizar esta acción corriendo dando un impulso hacia delante.
- ❖ **Sigilo:** Camina un 40% menos de la velocidad de caminar por defecto. Esta mecánica se desarrollará a partir del nivel 2.
- ❖ **Agacharse:** Esta acción solo está disponible cerca de cajas u objetos reduciendo su tamaño 40%. Esta mecánica se desarrollará a partir del nivel 2.

Interacciones:

- ❖ **Hablar:** Esta acción está disponible con los NPC's al posicionarse enfrente y si están quietos.

- ❖ **Agarrar-soltar:** "Inti" puede tomar props (Cajas, troncos cuadrados, llaves, piedras, Rocas y Ramas) del escenario y llevarlos consigo hasta que decida soltarlos.
- ❖ **Lanzar:** Disponible si el personaje ha tomado un objeto, esta acción se ejecuta al mantener pulsado el botón de agarrar luego de un segundo, lanza en dirección de donde el personaje está mirando.
- ❖ **Empujar:** Puede desplazar ciertos objetos del escenario caminando detrás de ellos.
- ❖ **Obtener objetos:** En ciertas ocasiones, después de conseguir completar una misión el personaje podrá obtener objetos como llaves, frutas o partes de un mapa.

Combate:

- ❖ **Disparar:** El jugador cuenta con **munición** (Piedras) base **limitada**, que lanza en dirección de donde está mirando el personaje destruyéndose al colisionar con muros, enemigos u objetos del escenario.
Cuando consigue la munición **piedras mágicas**, la munición de piedras se vuelve **ilimitada**.
- ❖ **Ataque melee:** Solo si dispone de una rama podrá realizar una serie de ataques que daña a los enemigos que están cerca y a rango del arma. Esta mecánica se desarrollará a partir del nivel 2.
- ❖ **Lanzar arma:** Disponible si el personaje cuenta con un arma arrojadiza (Ramas, Rocas o cajas). Esta mecánica se desarrollará a partir del nivel 2.
- ❖ **Esquivar:** Un desplazamiento hacia una dirección con una velocidad aumenta al triple de la velocidad de caminar. Esta mecánica se desarrollará a partir del nivel 2.

5.7.2 Combate

"Inti" dispone de disparo, lanzar armas (Armas arrojadizas), ataque cuerpo a cuerpo (solo con ramas o palos equipados) y esquivar.

- ❖ **Disparo:** Lanza piedras hacia delante donde esté mirando el personaje, como arma base su daño es un punto de vida al primer impacto con cualquier enemigo.
- ❖ **Ataque a melé:** Solo cuando el personaje ha conseguido un palo o rama, le permitirá realizar una serie de 3 ataques consecutivos, causando 2 puntos de daño al primer impacto con uno o varios enemigos.

- ❖ **Esquivar:** El personaje puede deslizarse hacia una dirección el triple de velocidad base, esta habilidad tiene un cooldown de 1 segundo aproximadamente. Esta mecánica se desarrollará a partir del nivel 2.
- ❖ **Lanzar objetos:** Solo si el jugador ha agarrado una caja este podría lanzarla hacia adelante, impactando a los enemigos con 2 puntos de daño al primer impacto.

5.7.3 Armas

"Inti" cuenta con 2 tipos de armas piedras y ramas cada una con posibles variantes ya sea de daño o aspecto.

- ❖ **Lanzar piedras:** Como munición infinita base, "Inti" puede disparar en dirección de donde está mirando ocasionando 1 punto de vida en base a los enemigos que se destruyen al primer impacto. Las variantes de esta arma (Piedra chispa causa 2 puntos de vida al primer impacto a uno o varios enemigos que se encuentren en la misma dirección con un máximo de 3 impactos antes de destruirse).
- ❖ **Ataque con palo:** "Inti" puede tomar ramas que se hayan caídas por el escenario para realizar ataques cuerpo a cuerpo con un total de 6 impactos antes de destruirse causando daño de 2 puntos de vida a los enemigos, Bastón de las edades es un arma indestructible muy pesada que causa 3 puntos de daño a la vida de los enemigos, pero tiene un cooldown, de 2 segundos aproximadamente. Esta mecánica se desarrollará a partir del nivel 2.

5.7.4 Vidas

"Inti" cuenta con una barra de vida que representa 10 puntos de vida y que se reduce al contacto con enemigos (-1) o al impacto de una de sus armas (-2 melé) y (-3 a las flechas o disparos).

Además, Inti cuenta con 3 vidas para poder superar el juego. Cada vez que la barra de vida queda reducida a cero se descuenta una vida del contador y se reinicia la barra de vida.

5.8 NPC'S (NATIVOS DE LA ISLA)

El juego tendrá varios NPC's, que comparten los mismo rasgos y cultura de nuestro personaje, pero los más importantes que intervienen en la historia serán descritos a continuación:

- ❖ **Anciano sabio:** No existe registro de su edad, pero se sospecha que ya rodea los 90 años, padre adoptivo de "Inti", al enterarse de la muerte de sus padres él decidió cuidar de nuestro héroe, al ser una persona mayor casi no puede ponerse en pie, pero siempre espera que "Inti" le visite en su casa al otro lado de la montaña, siempre tendrá un consejo para él.
- ❖ **Anciana sabia:** No existe registro de su edad, pero se sospecha que ya rodea los 90 años, padre adoptivo de "Inti", al enterarse de la muerte de sus padres él decidió cuidar de nuestro héroe, al ser una persona mayor casi no puede ponerse en pie, pero siempre espera que "Inti" le visite en su casa al otro lado de la montaña, siempre tendrá un consejo para él.
- ❖ **José:** Compañero de travesuras de "Inti", ¡cómplice! en casi todas sus hazañas, le admira mucho por ser el mayor y quiere seguir sus pasos, con apenas 8 años quiere seguir a "Inti" pero sus padres siempre le vigilan mucho, le gusta mucho el mar y suele estar por las orillas al este, si conversas con él siempre podrá poner al tanto de la situación de la isla.
- ❖ **Don Gustavo:** El tendero su lema es "La tienda siempre abierta", puedes acercarte a él para adquirir provisiones antes de partir a tu nueva aventura se encuentra en el centro de la isla.
- ❖ **George:** Una tortuga bebé de tipo galápagos el nombre de las islas se debe a esta especie de tortuga que aparte de ser longevas también suelen crecer más de lo común. Una de las 5 crías bebés que habitan en el estanque tortuga, su nombre se lo debe a su amigo guardián del estanque "Inti". Luego de ser rescatado de los piratas, George consigue comunicarse con Inti para pedirle que salve a sus hermanos. Desde ahí será su compañero y voz guía para las siguientes aventuras

5.8.1 Comportamiento

Las mecánicas básicas de estos NPC's:

- ❖ Patrullaje.
- ❖ Hablar, interacción con personaje principal.

5.8.2 Combate

No hay previsto que estos NPC's participen en combates. Al iniciar un combate entre "Inti" y algún villano si hay un NPC's cerca, este se pondrá a buen recaudo, siendo que no participan en combate y tampoco pueden ser dañados.

5.8.3 Armas

No hay previsto que estos NPC's participen en combates y por lo tanto no disponen de armas.

5.8.4 Vidas

No hay previsto que estos NPC's puedan morir, por lo tanto, persistirán en el nivel mientras estén activos. Es posible que durante el desarrollo puedan cambiar de posición en el nivel como los elementos de decoración para dotar de vida el nivel.

5.9 ENEMIGOS – PIRATAS

Los piratas son los enemigos de "Inti" que buscan arrebatarse y llevarse las especies fuera de la isla, intentarán seguirte y atacarte hasta que te derriben con sus golpes.

5.9.1 Comportamiento

Las mecánicas básicas de los enemigos serán:

- ❖ **Vigilancia:** Los enemigos se detienen por un tiempo en caso de no interceptar al jugador.
- ❖ **Patrullaje:** En caso de no detectar al jugador recorrerá los puntos de patrullaje para recorrer.
- ❖ **Correr:** En caso de detectar al player se desplaza más rápido hasta alcanzarlo.
- ❖ **Atacar:** Al conseguir alcanzar el player se realiza la animación de ataque.
- ❖ **IA de ataque agresivo:** en ocasiones los piratas tendrán una IA que les obligue a ir directamente a la posición del jugador.

5.9.2 Combate

Los enemigos son de dos tipos:

- ❖ **Melé:** Los enemigos al detectar al player mediante un rango de visión corren hacia el jugador y al momento de estar a una distancia prudente, tratarán de dañar, en caso de no estar a la vista del jugador estos procederán a patrullar.
- ❖ **Distancia:** Con arcos o con resortera en mano son más débiles que los enemigos a melé, pero mucho más letales, de manera estática, pero con más rango de visión son capaces de detectar y atacar a distancia, casi siempre se les ve acompañados de un enemigo a melé para evitar que se les puedan acercar.

5.9.3 Armas

- ❖ **Melé - garfio:** Algunos piratas disponen de un garfio de corto alcance de medio metro capaz de causar 2 puntos de daño a la vida del personaje.
- ❖ **Melé - Espada:** Una espada de largo alcance que agita haciendo una suerte de estocada con un rango de alcance de 3 metros causa 3 puntos de daño a la vida del jugador.
- ❖ **Distancia - Resortera (Tirachinas):** Capaz de lanzar piedras a una distancia de 10 metros con 1 punto de daño a la vida base del héroe y con un tiempo de recarga de 2 segundos.
- ❖ **Distancia - flecha:** Con una distancia de 30 metros de alcance y con un daño de 4 puntos de vida son un arma letal. Pero que se puede esquivar y con un tiempo de recarga de 3 segundos.

5.9.4 Vidas

Los enemigos a melé cuentan con un total 20 puntos de vida mientras que los enemigos a distancia 8 puntos de vida.

5.10 OTRAS MECÁNICAS

5.10.1 Cajas con especies para rescatar

En diferentes espacios del juego aparecerán cajas (cubos) representando diferentes especies de la isla que los piratas han atrapado. Una de las dinámicas del juego consiste en rescatar especies, por tal razón, mediante las mecánicas de coger, soltar y lanzar estas especies podrán ser liberadas ya sea porque se traslada la caja de un punto seguro habilitado como tal en el

escenario o bien porque al colisionar contra un pirata la caja se destruye y la especie queda liberada de su jaula.

5.10.1.1 Sistema de puntuación

A continuación, se describe cual es la lógica para la implementación del sistema de puntuación:

MISION 1, el objetivo es rescatar especies, en concreto 5 de cada una de las presentadas (IGUANAS, LEONES Y TORTUGAS), existe una variable temporal (cuanto vas a tardar en conseguirlo) y una variable vida (cuantas vidas tienes o cuanta vida te queda cuando finalices), también existe otra variable que es los ENEMIGOS, cada ENEMIGO puede ser destruido o eliminado lanzando una caja.

CADA CAJA DE LEONES SON 5 PUNTOS

CADA CAJA DE IGUANAS SON 10 PUNTOS

CADA CAJA DE TORTUGAS SON 15 PUNTOS

CADA ENEMIGO ELIMINADO SON 20 PUNTOS

El sistema de puntuación se calcula al final de la misión con la siguiente formula:

$$\text{PUNTUACION} = (\text{CAJA 1} + \text{CAJA 2} + \text{CAJA 3} + \text{ENEMIGOS}) \times \text{VIDAS} \times 100 / \text{TIEMPO}$$

MISION 2, el objetivo es rescatar a la tortuga George, el sistema de puntuación en este nivel se implementa de la siguiente forma:

$$\text{PUNTUACION} = \text{ENEMIGOS DERROTADOS} \times \text{VIDA} \times 100 / \text{TIEMPO}$$

MISION 3, el objetivo es llegar al final del nivel habiendo recogido las 10 piezas del mapa que completan el camino hacia la FUENTE MAGICA.

Por el camino, nos encontramos con obstáculos, el sistema de puntuación es el siguiente:

CADA MONEDA DA 1 PUNTO

CADA PUERTA DA 2 PUNTOS

CADA MURO DESTRUCTIBLE DA 5 PUNTOS

CADA OBSTACULO SALTADO DA 5 PUNTOS

CADA MURO MOVIL DA 10 PUNTOS

CADA PIEZA DEL MAPA DA 25 PUNTOS

El sistema de puntuación final se calcula de la siguiente forma:

$$\text{PUNTUACION} = (\text{MONEDAS} + \text{PUERTA} + \text{MUROS} + \text{SALTOS} + \text{PIEZA}) \times \text{VIDA} \times 100 / \text{TIEMPO}$$

5.10.1.2 Spawn de Cajas

La aparición de las cajas en el sistema es mediante un blueprint Generador de Cajas (spawn) que se colocarán estratégicamente en el escenario.

5.10.1.3 Cajas móviles

En alguna situación, el jugador deberá moverse en salas o espacios esquivando unas cajas móviles (movimiento de un lado a otro de la sala, de pared a pared, por ejemplo, de izquierda a derecha en línea recta, "rebotando" en la pared). Si uno de las cajas toca al jugador este deberá volver al punto de inicio de la sala.

5.10.2 Pulsadores

En el juego aparecerán pulsadores que actúan siempre de la misma forma, se trata de eventos "trigger" que generarán un resultado o bien al pulsarse de manera individual o conjunta y, en caso de manera conjunta, con un orden adecuado. Generalmente estos pulsadores van vinculados a un enigma que hay que resolver o de forma individual para eliminar un bloque o abrir una puerta.

5.10.2.1 Sistema de puntuación

No existe ningún sistema de puntuación asignado a esta mecánica que afecte directamente a la puntuación del jugador, sin embargo, la no resolución del enigma implica una limitación en el avance del juego o una penalización en el resultado del mismo.

5.10.2.2 Detalle de la mecánica para pulsadores individuales

Generalmente un pulsador que actúa de forma individual sirve como elemento para activar, abrir o desplazar un elemento como mecánica para acceder a una nueva sala. Es decir, generalmente actúan como elementos que abren puertas o hacen desaparecer bloques que obstaculizan la continuidad del nivel.

5.10.2.3 Detalle de la Mecánica de pulsadores conjuntos

Una pista o enigma definirá un orden correcto con el cual los pulsadores deben ser actuados, Así también, para superar el enigma, se deben pulsar determinados pulsadores con la imagen

representativa de especies en extinción en un orden concreto. Si una vez pulsado el orden no es correcto los pulsadores deben reiniciarse y el jugador deberá volver a pulsarlos, y con ello para establecer como pulsar un pulsador basta con colisionar contra él.

5.10.3 Bloques / Obstáculos

Otra de las mecánicas es la interacción con bloques, donde, en algunas zonas del mapa nos encontraremos con bloques que entorpecen el camino del protagonista y ahí dependiendo del tipo de bloque, la interacción del jugador será diferente y el premio y/o la penalización también.

5.10.3.1 Mecánica en Obstáculos/bloques básicos

Estos son bloques estáticos con los que en caso de que jugador colisiona con uno de ellos, perderá una vida. El bloque debe desaparecer una vez que el jugador colisiona con él. Existirán dos tipos de bloques, unos de poca altura que el jugador podrá esquivar saltando o moviéndose a derecha o izquierda y otros de más altura que no podrán ser saltados, por tal, no existe penalización por el contacto del jugador.

5.10.3.2 Mecánica en Obstáculos/bloques destruibles

Consiste en bloques que deben poder ser destruidos por el ataque del jugador y generalmente actúan como bloqueo de acceso a otras zonas y deben ser eliminados para facilitar el acceso a las nuevas zonas.

Es así que el contacto del jugador con ellos no es motivo de penalización ni de pérdida de vida, simplemente el jugador debe destruirlos para poder avanzar y tampoco generará ningún tipo de recompensa que afecte al sistema de puntuación, la recompensa en sí es poder avanzar en el nivel.

5.10.3.3 Mecánica en Obstáculos/bloques móviles

Este tipo de mecánica consiste en bloques que se mueven de forma automática de derecha a izquierda y posteriormente de izquierda a derecha (en forma de bucle), donde, el contacto del jugador con ellos no es motivo de penalización ni de pérdida de vida, simplemente el jugador debe esquivarlos para poder avanzar y tampoco tendrá ningún tipo de recompensa que afecte al sistema de puntuación, la recompensa en sí es poder avanzar en el nivel.

5.10.3.4 Mecánica en Obstáculos/puertas e interruptores

Existe un tipo especial de bloque de gran tamaño que actuarán como puerta los cuales deben desaparecer al pulsar un pulsador el mismo descrito en el apartado 5.10.2, el pulsador debe ser un elemento que se coloque en el juego antes de la posición de la puerta, cuando el jugador colisiona con el elemento pulsador la puerta debe desaparecer.

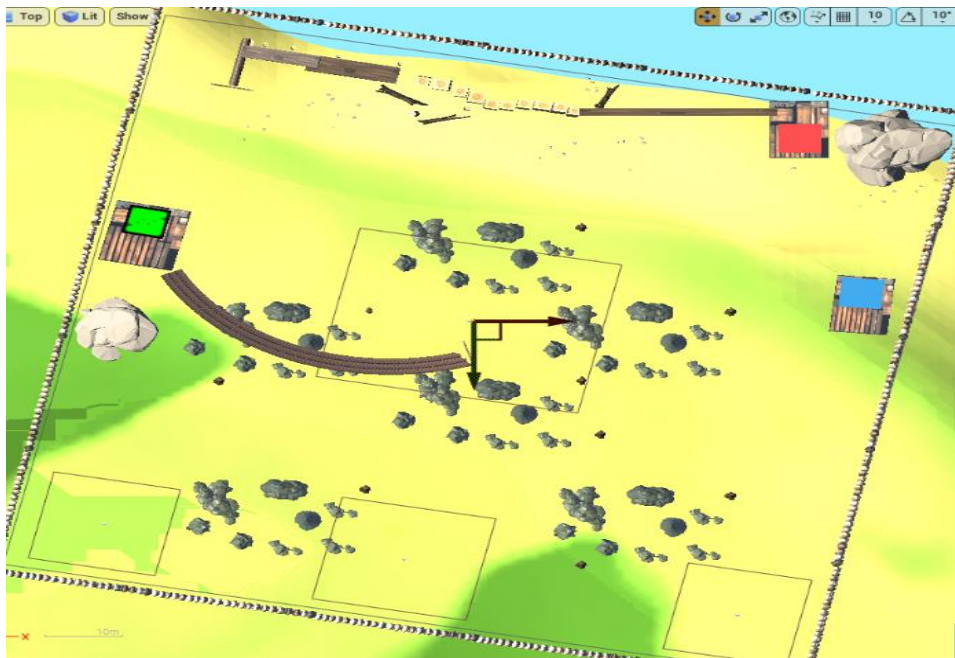
5.11 GAMEPLAY Y DINÁMICAS

El personaje principal “Inti” comienza en el sur de la isla, al avanzar se encuentra con un NPC que le indica la forma fácil de poder reconocer destinos, el uso de postes informativos que va a encontrarse por el camino, luego, le indica que, en una zona de la isla existen piratas que están capturando especies en extinción de la isla (tortugas, leones marinos e iguanas) y que debe acudir al rescate de las mismas.

El primer poste le indica que entre en una cueva en una roca creada por la erosión del agua, una vez dentro, el personaje se encuentra ante un entorno de laberinto donde hay un sistema de Spawn en modo aleatorio que crea cajas con las texturas de las 3 especies, también, existen 3 Spawn de piratas con un sistema que incrementa la dificultad con el tiempo.

Tras pasar por el teletransporte, desplazamos al personaje a un nivel nuevo, aunque aparentemente localizado en la misma isla, el escenario debe mantener la isla Isabela como trasfondo. Así, “Inti” se desplaza a esta parte de la isla donde se realizará la actividad de rescate de especies. El player inicia la partida en la posición central del escenario. Se detalla dicho nivel en la Figura 56:

Figura 56. Nivel 1, misión 1 rescate.



Fuente: Elaboración propia.

En el transcurso del juego en escenarios existen vallas exteriores que marcan el perímetro y obligan a limitar el área de participación del personaje, también, en la parte inferior se encuentran localizados 3 sistemas de SPAWN de piratas, el sistema de Spawn genera un nuevo pirata en cada punto cada 10 segundos, conforme pasa el tiempo se genera un pirata cada vez en menos tiempo hasta alcanzar los 2 segundos. En ese momento, el sistema de Spawn queda establecido y cada 2 segundos se genera un nuevo pirata.

La IA de los piratas es muy agresiva, van directamente contra el Player, así, conforme pasa el tiempo aparecerán más piratas, por lo que la misión se vuelve más frenética, más rápida con el sentido de enganchar más al jugador.

En la parte central, existe un sistema de Spawn de cajas, existen cajas de 3 tipos: tortugas, iguanas y leones marinos, es así, que el sistema de Spawn genera una caja aleatoria cada 3 segundos, de forma que con el tiempo cada vez hay más cajas en el espacio.

Por otra parte, los personajes de piratas mueren directamente al colisionar con las cajas y de la misma forma, las cajas se destruyen al colisionar con los piratas, esto genera la posibilidad de que el jugador pueda correr moviendo a los piratas para que ellos mismos colisionen con las cajas y se autodestruyan.

Existen 3 zonas de rescate de especies, ver Figura 57, marcadas con los cuadrados rojo, verde y azul, donde, la zona verde se rescatan las cajas de iguanas, en la zona azul se rescatan las cajas de leones marinos y en la zona roja se rescatan las tortugas, Así, cada una de las zonas de rescate quedan señalizadas con un HUB inmersivo:

Figura 57. Zonas de rescate de especies.



Fuente: Elaboración propia.

Cada caja rescatada acredita al jugador 1 punto, así, cada caja usada en destruir piratas no contabiliza y conforme aumente el tiempo de juego del jugador y se complique su juego, con la finalidad de atraer más su entusiasmo, se implementará un sistema que multiplique los puntos conseguidos en el rescate de cada especie, de forma que pueda terminar la misión rescatando solo 2 especies, se detalla acciones dentro del juego:

- 6 **Coger cajas:** Usando la tecla izquierda del ratón mientras Inti colisiona con las cajas, se cogerán las cajas.
- 7 **Lanzar las cajas:** Usando la tecla derecha del ratón Inti podrá lanzar las cajas
- 8 **Soltar las cajas:** Cuando "Inti" llega a las zonas de rescate, dejará caer las cajas empleando la tecla E.

En el acceso a las zonas de rescate, existen 3 niveles de dificultad:

- ❖ **Rescate de leones marinos:** Dificultad básica, el acceso a esta zona es sencillo, solo se trata de llegar a la zona, debe estar un poco elevada para evitar que los piratas puedan acceder a la misma, funcionará también como un punto de seguridad para el jugador.

Ver Figura 58:

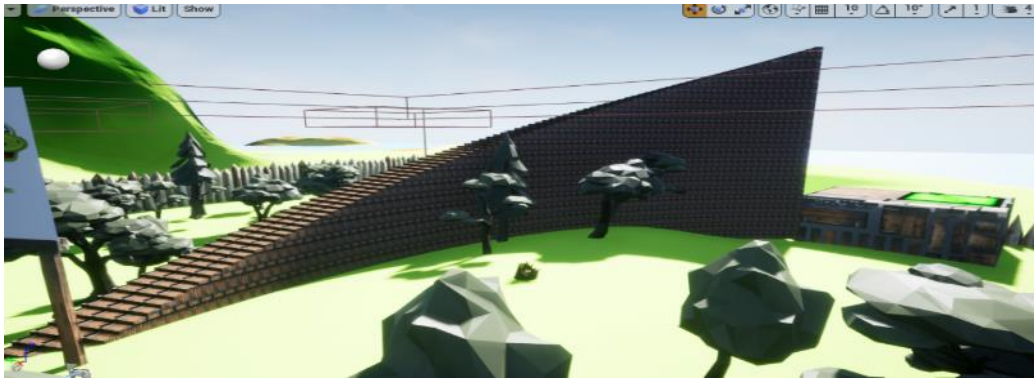
Figura 58. Zona de rescate de lobos marinos.



Fuente: Elaboración propia.

- ❖ **Rescate de iguanas:** Dificultad media, el acceso a la zona es a través de una gran escalera, al final de la misma, el jugador debe saltar al vacío para acceder a la zona de rescate. La dificultad radica en que pueden caer cajas que obstaculicen la subida y los piratas pueden acceder de forma que pueden perseguirnos en la ascensión. Ver Figura 59:

Figura 59. Zona de rescate iguanas.



Fuente: Elaboración propia.

- ❖ **Rescate de tortugas:** Dificultad alta, el usuario, transportando la caja, debe de desplazarse a través del circuito de troncos hasta llegar a la zona de rescate. Debido al tamaño de las cajas y el poco espacio de algunas zonas, es necesario que el jugador salte y se mueva para conseguir desplazar las cajas, generando espacios suficientes para acceder al siguiente escalón. Ver Figura 60:

Figura 60. Zona rescate de tortuga.



Fuente: Elaboración propia.

Como fin de la misión el objetivo es obtener 15 rescates a consecuencia de rescatar 5 especies de cada tipo, aunque, conforme el usuario se vaya cayendo en falta de avances, se permitirá contabilizar las cajas con mayor valor y por lo tanto podrá ir reduciendo el número de cajas a rescatar para poder conseguir la bolsa mágica que le permitirá almacenar rocas que usará posteriormente como munición en su siguiente desafío.

El personaje principal "Inti", al término de su primera misión no ha podido encontrar a su amigo George la tortuga, por tal razón, busca ayuda preguntando a los aldeanos del lugar, los aldeanos le comentan que los piratas han armado un fuerte y que es probable que el amigo

que busca este prisionero ahí, armado con su bolsa mágica y con la munición lista se debe dirigir hacerles frente a los captores de su amigo. Ver Figura 61:

Figura 61. Nivel 1, Acción 2 Inti.

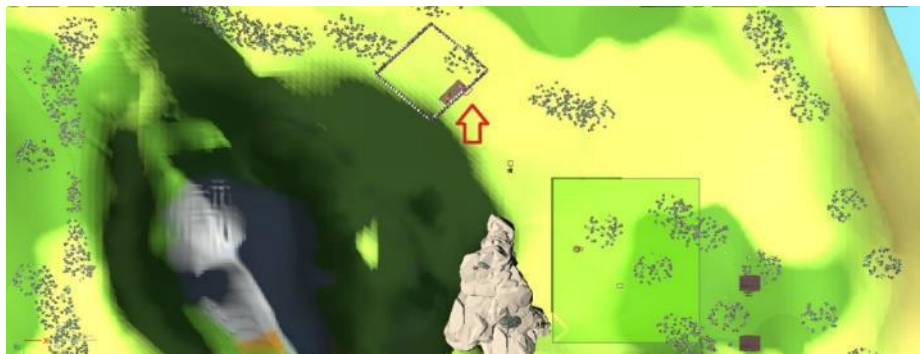


Fuente: Elaboración propia.

Como Inicio del juego, inti entra en el fuerte, y el sistema lo teletransporta a la nueva escena donde se ejecutará la misión dos, en la cual, dependerá del ingenio del jugador para poder sortear los obstáculos de cada uno de ellos. En el último nivel la vida del jugador corre peligro ya que los enemigos pueden causar daño reduciendo tu barra de vida con un máximo de 6 golpes bastara para derrotar al personaje.

Continuando con el fin del juego se termina cuando la barra de vida llega a 0, o cuando derrotas a todas las oleadas de los secuestradores. Ver Figura 62:

Figura 62. Inicio y fin de oleadas.



Fuente: Elaboración propia.

A continuación, se describe con más detalle el gameplay en esta misión 2:

- ❖ Nivel 1, misión 2, rescate de George

"Inti" entra al fuerte de los enemigos e investiga que ésta pasando: mensaje *"Ábrete paso sorteando los obstáculos, en este nivel deveras mover los bloques para poder acceder a zonas inaccesibles hasta encontrar el interruptor que abre la puerta, calcula el impulso del salto para evitar no caer"*. Para bajar nivel de dificultad pondrán guías en el suelo para que el jugador sepa la posición de los obstáculos.

Para completar el nivel deberás hacer uso de la fuerza empujando los bloques y calculando tus saltos, aplica las siguientes mecánicas:

- ❖ **Salto:** Se puede saltar sin impulso o corriendo esto te permitirá saltar a más distancia.
- ❖ **Empujar:** Caminar cerca de un bloque podrás empujarlo y en la dirección deseada.
- ❖ **Botón:** Si te sitúas sobre un botón estos abrirán una puerta (Por lo general la que más cerca se encuentre).

Figura 63. Nivel 1, misión 2 rescate.



Fuente: Elaboración propia.

En el Escenario del pantano, debe procura no caer en las plataformas que se mueven hasta poder cruzar el lago, una vez consiga llegar al otro extremo puede tomar la llave para abrir la puerta, "Inti" no sabe nadar así que cada vez que tocas el agua serás teletransportado al inicio del nivel. Para bajar el nivel de dificultad se bajará la velocidad de las plataformas móviles.

Se detalla las mecánicas que puede el jugador para desplazarte sobre las plataformas móviles y ser teletransportado al inicio del juego si te caes al agua y tomar la llave para abrir la puerta.

Ver Figura 64:

- ❖ **Viaje sobre plataformas:** Las plataformas se mueven cada una con diferente velocidad y diferente dirección.
- ❖ **Llave:** Al acercarte a una llave puedes tomarla y usarla para abrir la puerta.
- ❖ **Teletransporte:** Al caer al agua o simplemente tocarla te teletransportará al inicio del nivel.

Figura 64. Zona del pantano.



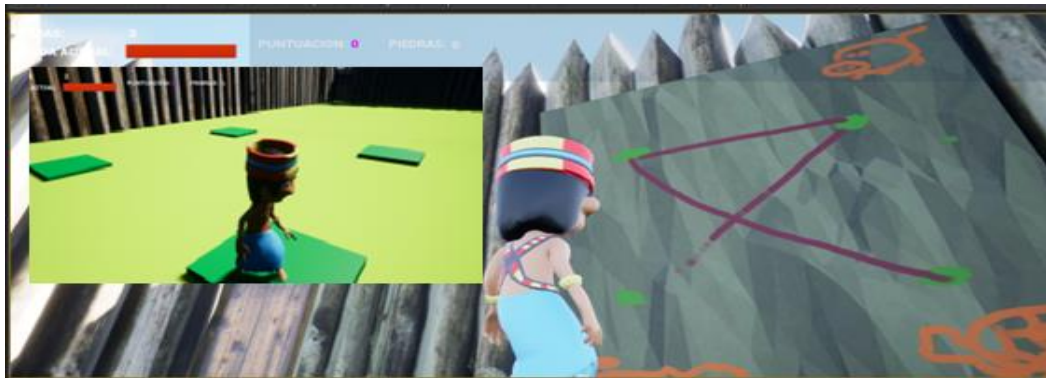
Fuente: Elaboración propia.

La opción de resolver un puzle para abrir la puerta de la celda estará activa, ejemplo alguien ha dejado unas pinturas antiguas en la pared, guíate de ella para resolver el acertijo que te permitirá abrir la puerta. Para bajar el nivel de dificultad tendrán la posibilidad de cometer más de dos errores antes de reiniciar los botones.

Para resolver el puzle se detallan las siguientes mecánicas, accionando los botones en el orden que sugiere la pintura rupestre. Ver Figura 65:

- ❖ **Botones en secuencias:** Si te posicionas sobre los botones de manera adecuada esto cambiarán de color verde en caso de que la secuencia sea equivocada estos se pondrán de un color rojizo se reiniciarán.
- ❖ **(Derrotar a los secuestradores.) - Combate:** Enfréntate a las oleadas de los enemigos hasta acabar con todos y así poder salvar a George la tortuga. Para reducir la complejidad del nivel se reducirá el límite de enemigos por oleada.
- ❖ **Disparo:** Si tienes buena puntería podrás derrotar a los enemigos, dispones de una barra de vida.

Figura 65. Resolver puzle.



Fuente: Elaboración propia.

❖ Nivel 1, misión 3

Al rescatar a George. Después de haber ganado la batalla y liberar a sus amigos Inti deberá dirigirse a una puerta donde empezará el gran escape poniendo a prueba sus habilidades y así salir sano y salvo del fuerte.

El jugador avanza hacia delante de forma constante y automática, también, puede mover al jugador a derecha e izquierda por lo que el objetivo del juego es avanzar hasta llegar a tocar el estanque, fin de nivel, esquivando todos los obstáculos que aparecen en el transcurso del nivel, una vez que llegue al estanque se consigue coger las piedras mágicas, en consecuencia, el gameplay descrito anteriormente obliga a la necesidad de desarrollar una serie de dinámicas alrededor del mismo

El inicio del juego empieza cuando "Inti" va corriendo hasta encontrar las piedras mágicas que se encuentran al final en un estanque mágico. En consecuencia, el juego termina una vez que el recorrido se complete en un determinado tiempo que transcurre el juego y que llegando a la fuente mágica antes que el BOSS.

5.11.1 Mecánicas del Nivel 1 – Misión 1 - Rescate de especies

La primera propuesta de juego tras hablar con los nativos, consiste en rescatar especies que están siendo capturadas por los piratas y para esta escena, el jugador se debe controlar con teclado (movimiento) y ratón (cámara), el objetivo del juego es recoger/transportar cajas de especies de diferentes colores que aparecen por el nivel de forma aleatoria y llevarlos a unas áreas específicas que habrá determinadas en el suelo.

Aparecerán múltiples cajas de forma aleatoria, de 3 especies diferentes, habrá también 3 pequeñas áreas específicas en el mapa sobre las que habrá que dejar las cajas de las especies y al depositar cada caja sobre su correspondiente área de rescate, se sumará un punto al marcador (solo se puntuará cuando el cubo y el área sean del mismo color).

En el nivel aparecerán diversos enemigos generados de forma automática que persiguen al jugador, estos enemigos podrán ser eliminados al lanzarles cajas, liberando también a las especies capturadas.

La funcionalidad para implementar será la siguiente:

- ❖ **Mapa:** Determinar el mapa de juego completo en el que se desarrollará el juego.
- ❖ **Cámara:** En modo tercera o primera persona, jugador controlable con teclado (movimiento) y ratón (vista).
- ❖ **Animaciones:** El jugador o el enemigo debe incluir una máquina de estados para gestionar al menos dos animaciones diferentes.
- ❖ **Carácter enemigo:** Los personajes controlados automáticamente que se dirigen hacia el jugador, poseen un ataque cuerpo a cuerpo que realizan al solaparse con el jugador. Este ataque debe reducir la vida del jugador hasta el punto de poder llegar a eliminarlo
- ❖ **Generación de enemigos:** Se deben incluir tres puntos diferentes de generación de enemigos en el mapa, se debe generar un enemigo cada N segundos, comenzando con $N=10$ y reduciéndose progresivamente según avanza el tiempo de juego, hasta llegar a un mínimo de $N = 2$.
- ❖ **Pawn cubos y generador de cubos:** Se deberán generar cubos aleatoriamente sobre el cielo y caer sobre el suelo, se generarán cubos cada 3 segundos, aunque según pase el tiempo se debe aumentar un número de cubos generados cada 4 segundos (comenzamos por uno hasta llegar a 4); Los cubos deben tener las propiedades físicas activadas (para que puedan ser transportados por el jugador), existirán cubos de 3 colores diferentes.
- ❖ **Acción recoger cubos:** Si se mantiene el botón izquierdo del ratón pulsado, cuando el jugador impacta contra un cubo se deberá recoger el cubo.

- ❖ **Acción lanzar cubos:** Si el jugador tiene un cubo debe poder lanzarlo pulsando el botón derecho del ratón, se lanzará el cubo aplicando un impulso hacia delante. Si un cubo lanzado impacta sobre un enemigo se deben eliminar ambos, el enemigo y el cubo.
- ❖ **Acción soltar cubos:** Si el jugador tiene un cubo debe poder soltarlo en el suelo (lentamente sin aplicar ningún impulso) cuando el usuario pulse tecla T.
- ❖ **Pawn áreas de colores:** Se deben incluir 3 áreas (superficies en el suelo) sobre las que el jugador debe depositar los cubos. Si el cubo que se deja en el área corresponde con el color de esta, el usuario obtendrá un punto.
- ❖ **Reglas de modo de juego:** El juego se ejecuta de forma infinita hasta que el jugador consiga rescatar 10 especies de cada o bien sea derrotado por los enemigos

5.11.2 Mecánicas del Nivel 1 – Misión 2 - Escape room

Después de rescatar las especies, un NPC sugiere a “Inti” que George la tortuga más importante de la isla, sigue capturada en una zona rodeada de enigmas, por lo cual, “Inti” acude una vez más a la acción.

Aquí, el jugador aparece en una sala delimitada por muros, la única posibilidad de pasar de una sala a otra es utilizando una puerta que inicialmente se encuentra cerrada. Para abrir la puerta el jugador debe resolver un puzzle, el juego consta de un total de cuatro salas contiguas (de una se pasa a otra). En la cuarta sala el jugador tendrá que luchar y destruir a varios enemigos para finalizar la prueba. En cada nivel encontrará un letrero que dará información sobre cuál es el reto que se debe superar en la sala de turno.

La funcionalidad para implementar es la siguiente:

- ❖ **Mapa:** Desarrollar el mapa de juego completo con las 4 salas en las que se desarrollará el juego. Recuerda que las salas deben estar conectadas entre sí mediante una puerta y que cada sala debe tener una estética visual /temática diferente.
- ❖ **Cámara:** La cámara en modo tercera o primera persona, jugador controlable con teclado (movimiento) y ratón (vista).
- ❖ **Animaciones:** El jugador o el enemigo debe incluir una máquina de estados para gestionar, al menos, dos animaciones.

- ❖ **Carácter enemigo:** Personajes controlados automáticamente que se dirigen hacia el jugador, poseen un ataque cuerpo a cuerpo que realizan al solaparse con el jugador. Este ataque debe reducir la vida del jugador hasta el punto de poder llegar a eliminarlo.
- ❖ **Plataformas móviles:** Se mueven en una sola dirección y a una velocidad configurable, una vez completado el recorrido estas plataformas se devuelven a su sitio original, esta acción la realizan de manera indefinida.
- ❖ **Generación de enemigos:** Los enemigos aparecen únicamente en la sala 4, cuando el jugador entre en la sala 4 se generarán 3 enemigos, con un intervalo de tiempo entre uno y otro de 4 segundos, y cada uno aparecerá dentro de la sala en una posición aleatoria.
- ❖ **Sala 1 - Empujar Bloques:** En esta primera sala deberá empujar bloques a un sitio en específico para usarlos como plataformas que le permitirán avanzar que mediante salto son inaccesibles, una vez posicionados en el sitio adecuado el jugador podrá acceder a un interruptor que al posicionarse sobre él abrirá la puerta a la siguiente sala.
- ❖ **Sala 2 - Plataformas móviles:** Para completar esta sala deberá saltar entre las plataformas sin caer en el agua ya que al caer el jugador será devuelto a la posición inicial, en este nivel se encuentran plataformas móviles que varían su destino y velocidad de movimiento las plataformas siempre que completan su destino se retornan repitiendo este ciclo indefinidamente, Para poder abrir la puerta a la siguiente sala es necesario encontrar una llave.
- ❖ **Sala 3 - Acertijo o Puzzle:** Para superar esta sala es necesario resolver el acertijo que se encuentra en la pared, son las indicaciones de la secuencia que se deben pulsar los botones del suelo, El mecanismo dispone de un feedback visual y auditivo al usuario.

Sala 4 – sala de rescate y combate: recordamos que cuando el jugador entre en la sala 4 se generarán 3 enemigos, con un intervalo de tiempo entre uno y otro de 4 segundos, y cada uno aparecerá dentro de la sala en una posición aleatoria. El jugador tiene solo una vida para completar el escape room.

5.11.3 Mecánicas del Nivel 1 – Misión 3 - Esquiva Obstáculo

Finalmente, al conseguir liberar a la tortuga George, esta nos sugiere que acudamos al estanque mágico a conseguir las piedras mágicas, un recurso infinito con el cual podemos atacar y derrotar más fácilmente a los piratas, sobre todo, al Boss.

La cámara enfoca al jugador desde el aire y desde atrás, el jugador avanza hacia delante de forma constante y automática, mientras, el usuario puede mover al jugador a derecha e izquierda utilizando las teclas A y D, en consecuencia, el objetivo del juego es avanzar hasta llegar a tocar el elemento de fin de nivel esquivando todos los obstáculos que aparecen en el transcurso del nivel.

La funcionalidad para terminar de implementar el juego es la siguiente:

- ❖ **Mapa:** Desarrollar el mapa completo del nivel por el que el jugador irá avanzando de forma automática. En algunas partes del nivel tiene que haber huecos por los que el jugador pueda caerse o cortes en el camino, si el jugador se cae el juego debe finalizar (pierde todas las vidas). Para detectar si el jugador se ha caído se puede colocar una superficie debajo del propio suelo del nivel, si el jugador impacta con esa superficie se finaliza el juego
- ❖ **Cámara:** La cámara en modo vista aérea, con las teclas A y D, el usuario debe poder mover al jugador hacia la derecha o la izquierda. El Jugador avanza hacia delante con velocidad constante.
- ❖ **Animaciones:** El jugador debe incluir una máquina de estados para gestionar al menos dos animaciones diferentes.
- ❖ **Character jugador:** Debe poder ser controlado con las teclas A y D (movimiento a derecha e izquierda), barra espaciadora para saltar y botón derecho del ratón para atacar (ataque cuerpo a cuerpo de corta distancia que se utilizará para destruir algunos tipos de obstáculos). El jugador debe contar con tres vidas.
- ❖ **Obstáculos/bloques básicos:** Se incluyeron muros estáticos, si el jugador colisiona con ellos, perderá una vida y el bloque desaparece. Existen 2 tipos de muros, uno que puede ser saltado con facilidad, y otro que definitivamente debe ser esquivado.

- ❖ **Obstáculos/bloques destruibles:** Los bloques de ladrillo quebradizo se pueden destruir con el ataque del jugador con el botón derecho del mouse, el jugador lanza una bola desde su mano derecha. Existen bloques de ladrillo amarillo que pueden ser destruidos, lanzando "piedras" con el botón derecho del ratón.
- ❖ **Obstáculos/bloques móviles:** Bloques de ladrillos anaranjados que se mueven de izquierda a derecha y viceversa en bucle, que provocan 1 daño de vida y son indestructibles.
- ❖ **Bloques/puertas e interruptores:** Existen bloques gigantes que bloquean el paso del jugador con un material tipo puerta, se destruye con el lanzamiento de la piedra con el botón derecho del mouse.
- ❖ **Objetivo, final del nivel:** Debido a que el nivel se genera dinámicamente, se decidió optar por una distancia recorrida objetiva, que a medida que avanza el jugador, la barra de progreso avanza, hasta llegar al final, donde el juego se da por terminado y este realiza una animación de festejo. Para ello el jugador debe ir recogiendo partes del mapa que se va encontrando por el camino.
- ❖ **Sumar vidas:** Cada vez que la recolecta de frutas llegue a ser múltiplo de 10, se sumará una vida. Música de vida extra.
- ❖ **Plataformas Huecas:** Se agregaron plataformas con un espacio donde el jugador se cae y pierde el juego automáticamente, esto no estaba implementado en la primera fase, pero en esta fase, sí.
- ❖ **Lanzar/ Ataque de jugador:** En esta fase se definió el ataque del jugador en la manera de lanzar piedras para destruir objetos y abrir puertas. Para poder lanzar piedras el jugador debe haberlas recogido previamente.
- ❖ **Para recorrido al morir:** Se implementó una lógica para para el contador al momento de morir, sea por perder todas las vidas o al momento de caerse en un hueco.

6 CONCLUSIONES

Las conclusiones de este trabajo se van desglosando en marco a los objetivos planteados, de los cuales se han ido cumpliendo y se obtenido como resultado el prototipo del videojuego llamado “Ancient Galápagos”, del cual se ha registrado los niveles, sus mecánicas y el GDD en este documento.

Por esta razón, el prototipo desarrollado incluye algunas mecánicas consideradas inicialmente como base y que en el transcurso de su implementación refleja un entorno escalable del cual se establecieron tiempos, se duplicó el esfuerzo y se fortaleció la dedicación de todo el equipo de trabajo para que el videojuego tenga un aspecto comercial y evolutivo.

Como línea futura de este trabajo se considera que la relación estrecha que tiene los videojuegos con las tecnologías son una ventaja para poder aplicar metodologías ágiles del cual su administración y control sobre el desarrollo del producto se vea enfocada de forma incremental e iterativa.

Así, para poder completar este proyecto antes los obstáculos de creación de animaciones, diseños 3D y la exclusión temprana de un miembro del equipo permitió establecer modificaciones ambiciosas por el equipo de trabajo siempre con el afán del aprendizaje continuo y el establecer técnicas de la metodología ágil para encontrar la cooperación, organización y la planificación que encamino al éxito de prototipo.

7 POSTMORTEN DEL JUEGO

Durante el desarrollo del prototipo el equipo de trabajo de este proyecto realizó una reunión final POSTMORTEN donde cada uno expuso los problemas o inconvenientes que aparecieron en el camino del desarrollo del videojuego y así también aportamos soluciones enfocados a que la experiencia adquirida sea un aprendizaje de superación.

7.1 FALTA DE EXPERIENCIA EN PROGRAMACIÓN UNREAL

Al escoger trabajar con Unreal Engine sabíamos sobre la falta de experiencia en su uso lo cual nos limitó y de cierto modo retrasó las tareas expuestas para el equipo, por lo cual, nos obligó a experimentar continuamente y auto educarnos viendo tutoriales, leyendo para desarrollar código, aunque en ocasiones no funcionaba.

Entendemos que, en un proyecto con un alcance y objetivo distinto nos tiene que llevar a trabajar en un entorno donde el equipo involucrado tenga conocimientos ya planteados de experiencia o de lo contrario el proyecto tiende a desarrollarse en un tiempo más largo de lo propuesto.

7.2 FALTA DE PERSONAL QUE INTEGRE EL EQUIPO

Al trabajar con la herramienta TRELLO nos permitió gestionar el proyecto y a su vez controlar las tareas implicadas para cada miembro del equipo, sin embargo, tuvimos una reacción tardía ante la exclusión de un compañero, la cual se intentó cubrir al principio con esfuerzo de cada integrante que quedaba, pero se comenzaron a generar cuellos de botella en el proyecto que afectaron su continuidad.

Lo positivo, mediante la aplicación de la metodología ágil nos enfocamos en poder darle un giro al proyecto comenzando por realizar pequeñas reuniones que facilitaban la comunicación para adaptar de manera flexible el trabajo a las circunstancias específicas de todo el entorno.

7.3 EXPERIENCIA DE USUARIO

El valioso aporte del guía y tutor de este trabajo quien supo apoyarnos fundamentando ideas en base a los diseños por niveles, en la exposición de comprensión del personaje principal, por tal razón, nuestro trabajo comenzó en la comprobación de niveles jugable y superable,

delimitaciones del incremento de dificultad y el comprender las exigencias del juego para conseguir que el jugador sienta el superar cada nivel con dificultades del juego.

BIBLIOGRAFIA

Chandler, H. M. (2020). *The Game Production Toolbox* (1st ed.). Taylor & Francis Group.

doi:10.1201/9780429440021

Computer Hope. (03 de 2022). *Third-person view*. Obtenido de

<https://www.computerhope.com/jargon/t/third-person-view.htm>

ISOMETRICORP. (2022). *TUNIC*. Obtenido de

<https://store.steampowered.com/app/553420/TUNIC/>

Keith, C. (2010). Addison-Wesley Professional. Obtenido de

https://www.goodreads.com/book/show/8442251-agile-game-development-with-scrum?from_search=true&from_srp=true&qid=1Af4Xp9SMU&rank=1

Maxwell, C. H. (2010). *The Game Production Handbook* (Segunda ed.). Jones & Bartlett

Learning. Obtenido de <https://www.oreilly.com/library/view/the-game-production/9781934015407/>

MobyGames. (2001). *SnowCraft*. Obtenido de

<https://www.mobygames.com/game/snowcraft>

The Legend of Zelda Wiki. (2002). Obtenido de The legend of Zelda: The Wind Waker:

https://zelda.fandom.com/es/wiki/The_Legend_of_Zelda:_The_Wind_Waker

UnrealEngine. (s.f.). *Epic Games*. Obtenido de [https://docs.unrealengine.com/4.27/en-](https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/ProgrammingWithCPP/UnrealArchitecture/)

[US/ProgrammingAndScripting/ProgrammingWithCPP/UnrealArchitecture/](https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/ProgrammingWithCPP/UnrealArchitecture/)