

BROKENLAND

Alex González Inglés
Máster Universitario
Diseño y programación de videojuegos

Helio Tejedor Navarro
Joan Arnedo Moreno

04/06/2020



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>BrokenLand</i>
Nombre del autor:	<i>Alex Gonzalez Ingles</i>
Nombre del consultor/a:	<i>Helio Tejedor Navarro</i>
Nombre del PRA:	<i>Joan Arnedo Moreno</i>
Fecha de entrega (mm/aaaa):	06/2020
Titulación:	<i>Máster Universitario</i>
Área del Trabajo Final:	<i>Desarrollo y programación de videojuegos</i>
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>Supervivencia, Fragmentado, LowPoly</i>
Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i>	
<p>El objetivo de este trabajo de final de máster es aplicar los conocimientos adquiridos a lo largo del máster, en todas las áreas de desarrollo de videojuego, para la creación de un videojuego desde cero. Se ha decidido la creación de un juego de supervivencia, por su gran popularidad y por ser un género no trabajado durante el curso, de esta forma, el alumno ha querido abordarlo como un reto. BrokenLands es un juego que transcurre en un momento posterior al actual, el cual, después de un fenómeno desconocido, el mundo se fragmenta. El jugador se encontrará con un mundo fragmentado en el cual deberá sobrevivir hasta encontrar una brecha que le permita regresar a su tiempo.</p> <p>La metodología utilizada para el desarrollo del juego ha sido incremental, creando pequeñas versiones previas a la final, con tal de ir añadiendo funcionalidades y mecánicas.</p> <p>El resultado es un juego completo, con un apartado visual atractivo para los jugadores, que puede ser ejecutado en máquinas de bajo rendimiento, debido a su baja carga gráfica. El juego consta de tres islas a modo de primer nivel del juego, con mecánicas complejas como la creación de construcciones, o la supervivencia del jugador.</p> <p>El alumno ha conseguido sus objetivos pese a la limitación de tiempo, ofreciendo un producto robusto, muy testeado y disfrutable por todo tipo de jugadores.</p>	

Abstract (in English, 250 words or less):

The objective of this final master's project is to apply the knowledge acquired throughout the master's degree, in all areas of video game development, to create a video game from scratch. The creation of a survival game has been decided, due to its great popularity and for being a genre not worked during the course, in this way, the student wanted to approach it as a challenge. BrokenLands is a game that takes place at a later time than today, which, after an unknown phenomenon, the world fragments. The player will encounter a fragmented world in which he must survive until he finds a gap that allows him to return to his time.

The methodology used for the development of the game has been incremental, creating small versions prior to the final version, in order to gradually add functionalities and mechanics.

The result is a complete game, with an attractive visual graphics for the players, that can be executed on low-performance machines, due to its low graphic load. The game consists of three islands as a first level of the game, with complex mechanics such as creating constructions, or player survival.

The student has achieved his goals despite the time limitation, offering a robust product, highly tested and enjoyable by all types of players.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	1
1.3 Enfoque y método seguido.....	3
1.4 Planificación del Trabajo.....	3
1.5 Breve resumen de productos obtenidos.....	7
1.6 Breve descripción de los otros capítulos de la memoria.....	7
2. Estado del arte.....	8
2.1 Género de juego.....	8
2.2 Tecnologías usadas.....	11
3. Definición del juego.....	11
3.1 Descripción.....	11
3.2 Tipo de interacción juego-jugador.....	12
3.3 Plataforma de destino.....	12
3.4 Historia y ambientación.....	12
3.5 Definición de los personajes y elementos.....	13
3.6 Interacción entre los actores del juego.....	13
3.7 Objetivos planteados al jugador.....	14
3.8 Concept Art.....	14
4. Diseño técnico.....	15
4.1 Entorno.....	15
4.2 Requerimientos técnicos del entorno.....	15
4.3 Herramientas empleadas.....	16
4.4 Recursos utilizados.....	16
4.5 Escenas/Pantallas.....	23
4.6 Diagrama de componentes.....	25
4.7 Global.....	26
4.8 Personaje.....	36
4.9 Acciones.....	38
4.10 Construcción.....	41
4.11 HUD.....	43
4.12 Entorno.....	45
4.13 Recursos.....	49
4.14 NPC e IA.....	54
4.15 Descripción de clases.....	62
5. Testing.....	65
6. Diseño de niveles.....	67
6.1 Mundo fragmentado.....	67
6.2 Isla inicial.....	67
6.3 Segunda isla.....	68
6.4 Isla destino.....	70
7. Manual del usuario.....	71
7.1 Requerimientos técnicos.....	71
7.2 Controles.....	71
7.3 Objetivos.....	72

7.4 Menús.....	72
7.5 Enemigos	73
8. Trabajo futuro	74
9. Conclusiones.....	75
10. Glosario	78
11. Bibliografía y Ludografía.....	79
11.1 Bibliografía	79
11.2 Ludografía	79

Lista de figuras

Ilustración 1 Gantt Semanas 1-5 1	4
Ilustración 2 Gantt Semanas 1-5 2	4
Ilustración 3 Gantt Semanas 1-5 3	5
Ilustración 4 Gantt Semanas 6-12 1	5
Ilustración 5 Gantt Semanas 6-12 2	6
Ilustración 6 Gantt Semanas 6-12 3	6
Ilustración 7 Gantt Semanas 6-12 4	6
Ilustración 8 Gantt Semanas 13-15	7
Ilustración 9 Minecraft	9
Ilustración 10 Ylands	9
Ilustración 11 Conan Exiles	10
Ilustración 12 The Long Dark	10
Ilustración 13 Estilo LowPoly de los paquetes Polygon	14
Ilustración 14 Ejemplo de isla flotante	14
Ilustración 15 Llanuras Quebradas	15
Ilustración 16 Polygon Adventure Pack	17
Ilustración 17 Polygon Vikings Pack	18
Ilustración 18 Polygon Prototype Pack	18
Ilustración 19 Cartoon FX Free	18
Ilustración 20 Farm Animal Pack	19
Ilustración 21 Easy Enemy Pack	19
Ilustración 22 Food Pack	19
Ilustración 23 PowerUpds	20
Ilustración 24 Space Pack	20
Ilustración 25 Basic Locomotion Free	20
Ilustración 26 Luna	21
Ilustración 27 Sol	21
Ilustración 28 Estrella	21
Ilustración 29 BrokenGlass Font	22
Ilustración 30 Underdog Font	22
Ilustración 31 Menú Principal	23
Ilustración 32 Controles	24
Ilustración 33 Opciones	24
Ilustración 34 Introducción	24
Ilustración 35 Escena de juego	24
Ilustración 36 Menú de pausa	24
Ilustración 37 Pantalla final	25
Ilustración 38 Diagrama de componentes	25
Ilustración 39 Guía 1	27
Ilustración 40 Guía 2	28
Ilustración 41 Guía 3	28
Ilustración 42 Guía 4	28
Ilustración 43 Estado no hambriento	29
Ilustración 44 Estado hambriento	29
Ilustración 45 Ciclo de supervivencia	29
Ilustración 46 Diagrama de supervivencia	30

Ilustración 47 Objetos sol y luna	31
Ilustración 48 Diagrama de rotación del sol	31
Ilustración 49 Amanecer	31
Ilustración 50 Día	31
Ilustración 51 Anochecer	32
Ilustración 52 Noche	32
Ilustración 53 Plataformas	32
Ilustración 54 Brecha	32
Ilustración 55 El jugador ha ganado	33
Ilustración 56 El jugador ha perdido	33
Ilustración 57 Muerte	33
Ilustración 58 Caída al vacío	34
Ilustración 59 Menú de pausa	34
Ilustración 60 Menú de inventario	35
Ilustración 61 Menú de crafting	35
Ilustración 62 Inventory Info	36
Ilustración 63 Colliders del avatar	36
Ilustración 64 Diagrama de animaciones de las acciones	37
Ilustración 65 Efecto picar una piedra	38
Ilustración 66 Efecto talar un árbol	38
Ilustración 67 Efecto de ataque	38
Ilustración 68 Acción coger	39
Ilustración 69 Acción cortar	39
Ilustración 70 Acción picar	39
Ilustración 71 Acción ataque	39
Ilustración 72 Paso 1: Coger el objeto	40
Ilustración 73 Paso 2-Inventario: Click en el objeto a comer	40
Ilustración 74 Paso 2-Acceso Rápido: Pulsar tecla asociada	40
Ilustración 75 Paso 3: Comer	40
Ilustración 76 Menú de crafting con indicadores	41
Ilustración 77 Sombra al construir	41
Ilustración 78 Construcción imposible	42
Ilustración 79 Hoguera construida	42
Ilustración 80 Hoguera	42
Ilustración 81 Cama construida	43
Ilustración 82 Jugador durmiendo	43
Ilustración 83 Corazones indicando la vida	43
Ilustración 84 Muslos de pollo representando el hambre	43
Ilustración 85 Indicador visual del sol	44
Ilustración 86 Indicador visual de la luna	44
Ilustración 87 Cinturón de herramientas	44
Ilustración 88 Acceso rápido	44
Ilustración 89 Isla inicial 1	45
Ilustración 90 Isla Inicial 2	46
Ilustración 91 Segunda Isla 1	46
Ilustración 92 Segunda Isla 2	47
Ilustración 93 Isla Destino 1	47
Ilustración 94 Isla Destino 2	48
Ilustración 96 Indicador de puente	48
Ilustración 97 Indicador flotante de puente	48

Ilustración 98 Puente construido 1	49
Ilustración 99 Puente construido 2	49
Ilustración 100 Árbol Isla Inicial	50
Ilustración 101 Árbol Segunda Isla	50
Ilustración 102 Árbol Isla Destino	50
Ilustración 103 Mena de piedra	50
Ilustración 104 Arbusto de bayas	51
Ilustración 105 Setas	51
Ilustración 106 Manzanero	52
Ilustración 107 Caída de fruta	52
Ilustración 108 Múltiples caídas de fruta	53
Ilustración 109 Manzanero talado	54
Ilustración 110 Llama	55
Ilustración 111 Diagrama IA Llama	55
Ilustración 112 Zonas de patrulla de Llama	55
Ilustración 113 Llama descansando	56
Ilustración 114 Llama caminando	56
Ilustración 115 Área de detección de serpiente	57
Ilustración 116 Diagrama IA Serpiente	57
Ilustración 117 Serpiente patrullando	58
Ilustración 118 Serpiente en alerta	58
Ilustración 119 Serpiente atacando	58
Ilustración 120 Diagrama IA Caballo	59
Ilustración 121 Áreas de patrulla de caballo	59
Ilustración 122 Caballo caminando	60
Ilustración 123 Caballo descansando	60
Ilustración 124 Área de detección de araña	61
Ilustración 125 Diagrama IA Araña	61
Ilustración 126 Araña patrullando	62
Ilustración 127 Araña atacando	62
Ilustración 128 Mundo de BrokenLand	67
Ilustración 130 Mapa Isla Inicial 1	68
Ilustración 131 Mapa Isla Inicial 2	68
Ilustración 133 Mapa Segunda Isla 1	69
Ilustración 134 Mapa Segunda Isla 2	69
Ilustración 136 Mapa Isla Destino 1	70
Ilustración 137 Mapa Isla Destino 2	71
Ilustración 138 Menú de controles	72
Ilustración 139 Menú de dificultad	72
Ilustración 140 Menú de pausa	72
Ilustración 141 Menú de inventario	73
Ilustración 142 Menú de crafting	73

1. Introducción

1.1 Contexto y justificación del Trabajo

En este trabajo de fin de máster, el alumno debe diseñar y desarrollar un juego desde cero; partiendo de los conocimientos previos adquiridos durante el curso. De esta manera, el alumno ha probado diferentes géneros y ha revisado cual puede ser totalmente diferente a los géneros ya estudiados en el máster. De esta manera, este trabajo se ha abordado desde el enfoque de reto, debido a que el género y las mecánicas que tiene un juego de este tipo, no se habían explicado ni desarrollado nunca durante el curso.

El género *survival* está en auge y cada vez más desarrolladores optan por este género, debido a dos factores:

- Libertad: Ofrece al jugador multitud de posibilidades.
- Creatividad: Al haber un vasto mundo que el jugador debe explorar, el desarrollador puede crear lo que quiera, debido a que, al ser un mundo libre, puede crear cualquier cosa.

El alumno se ha basado en las preferencias actuales del mercado de videojuegos y en experiencia personal con el género, para poder crear un juego de supervivencia con elementos diversos, presentes en el género.

1.2 Objetivos del Trabajo

El objetivo de este trabajo de final de máster es la creación de un videojuego de temática *survival* ambientado en un mundo fragmentado. Este videojuego se debe diseñar de tal forma que tenga un objetivo final, para poder dar por finalizado el juego, en el caso de BrokenLand, llegar a la brecha temporal.

Los objetivos se han dividido por áreas, para así unificar desarrollos de una misma área y tener conciencia de a qué área hay que dedicarle más esfuerzo.

1. General

- a. Menú inicio: Para iniciar la partida o salir.
- b. Menú de pausa: Pausar el juego en todo momento para salir del juego.
- c. Menú de inventario: Abre el inventario del jugador para saber qué elementos tiene encima.
- d. Mecánica y pantalla de final del juego
- e. Pequeño mensaje narrativo/imagen al inicio para introducir al jugador

2. HUD

- a. Vida
- b. Hambre
- c. Hora / Momento del día

3. Personaje

- a. Modelo
- b. Movimiento
- c. Animaciones movimiento

4. Acciones del personaje
 - a. Ataque
 - b. Tala
 - c. Picar
 - d. Recoger
 - e. Comer
5. Construcciones
 - a. Sistema general de construcción
 - b. Hoguera
 - c. Cama
 - d. Puente
6. NPC
 - a. Elemento vivo pacífico 1
 - b. Elemento vivo pacífico 2
 - c. Elemento vivo agresivo 1
 - d. Elemento vivo agresivo 2
7. Inteligencia Artificial
 - a. Pathfinding
 - b. Máquina de estados
 - c. Mecánica agresiva
 - d. Mecánica pasiva, huir
 - e. Ligera diferencia entre todos los elementos vivos, ya sea la vida, o cambios de comportamiento específicos
8. Mecánicas globales
 - a. Regeneración de árboles
 - b. Regeneración de bayas
 - c. Regeneración de menas
 - d. Caída de fruta de los árboles
 - e. Mecánica de hambre / regeneración de vida
9. Recursos
 - a. Madera: tala de árboles
 - b. Metal / Piedra: Picar una mena
 - c. Bayas: Recolectar de arbusto
10. Diseño de niveles
 - a. Isla inicial
 - b. Árboles
 - c. Menas
 - d. Arbustos
 - e. Mecánica de conexión de islas con puentes
 - f. Segunda isla
 - g. Isla destino
11. Correcciones
 - a. Bugs del juego
 - b. Bugs de la IA
12. Tests
 - a. Tests de IT
 - b. Tests con usuarios

13. Documentación / Entregables

- a. Informe Versión Parcial (PEC 2)
- b. Vídeo Versión Parcial (PEC 2)
- c. Informe Versión Jugable (PEC 3)
- d. Vídeo Versión Jugable (PEC 3)
- e. Memoria TFM
- f. Vídeo explicativo TFM
- g. Autoevaluación

1.3 Enfoque y método seguido

El trabajo de final de máster se ha dividido en diversas entregas, de esta manera, el alumno ha podido centrar los esfuerzos para finalizar cada entrega.

El alumno se ha centrado en una metodología ágil, basada en tareas:

- Todas las tareas se han listado al inicio del proyecto.
- Se han distribuido las tareas en las diferentes entregas del trabajo final.
- Cada tarea es autocontenida, eso significa que se ha de finalizar el desarrollo de esa parte para darla por cerrada. (Por ejemplo, no se ha abordado la creación de recursos nuevos, hasta la creación los recursos básicos (madera y piedra)).

Cada isla dentro del juego se ha programado como entidades separadas, de esta forma se probaba por separado cada isla, para posteriormente unificar todos los elementos en el flujo de juego. El desarrollo ha pasado por varias fases, que representan cada isla del juego.

Una vez cada isla tenía los elementos adecuados, se procedía a probarlos y si estaban correctos, se integraban en el juego base.

Esta metodología de desarrollo, favorece la creación de objetos finalizados, en ningún caso (más allá de bugfix) se ha dejado algún componente a medias. Primero, se han finalizado los objetos de forma individual y luego se han unificado.

1.4 Planificación del Trabajo

Dentro del trabajo de final de máster existen diversas entregas y por esta razón se ha dividido el trabajo para tener diferentes versiones del juego. Se ha diseñado un diagrama de Gantt para poder tener un control del tiempo estimado de cada tarea y luego modificarlo para ver si al final del proyecto se han cumplido la estimación de las tareas y los plazos. Cada tarea ha sido identificada como una unión de número y letra y se puede ver en el punto [1.2](#) su descripción. El diagrama está separado por semanas para facilitar su comprensión.

La planificación inicial se ha cumplido, pero se ha extendido la fase de testing para que englobe todo el desarrollo. Se han realizado los tests a medida que se ha ido desarrollando.

Las semanas de entregas han sido resaltadas con naranja, para que quede bien claro.

Versión Parcial – PEC 2 – 02/03/2020 a 05/04/2020

En esta primera versión del juego no se busca que el jugador tenga una experiencia real del juego, sino una pequeña demo del primer nivel del mismo y con funciones limitadas. Se definió una zona alejada del inicio para simular el final de juego y ver que ocurre.

Se ha diseñado la primera isla del juego con sus elementos, aunque el diseño no fue el definitivo, ya da un ejemplo de que podemos encontrarnos en cada isla. También permite al jugador mover su avatar libremente y realizar algunas acciones básicas para interactuar con el mundo y hacerse una idea de las mecánicas del juego.

También el jugador tiene a su alcance los diferentes recursos que se encontrará en la versión final del título.

Un HUD, para nada definitivo, ayuda al jugador a saber que parámetros deberá cuidar en la versión final del juego.

Por último, se realizó el informe correspondiente a esta versión del juego y un breve vídeo explicativo.

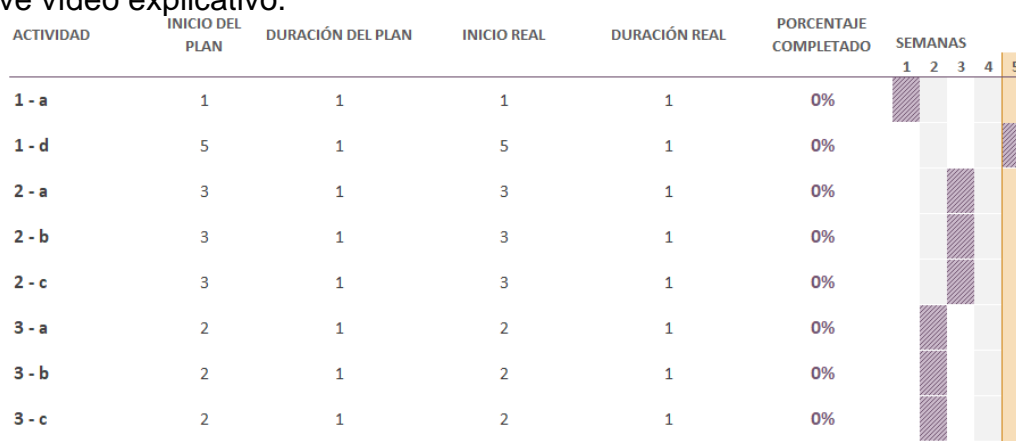


Ilustración 1 Gantt Semanas 1-5 1



Ilustración 2 Gantt Semanas 1-5 2

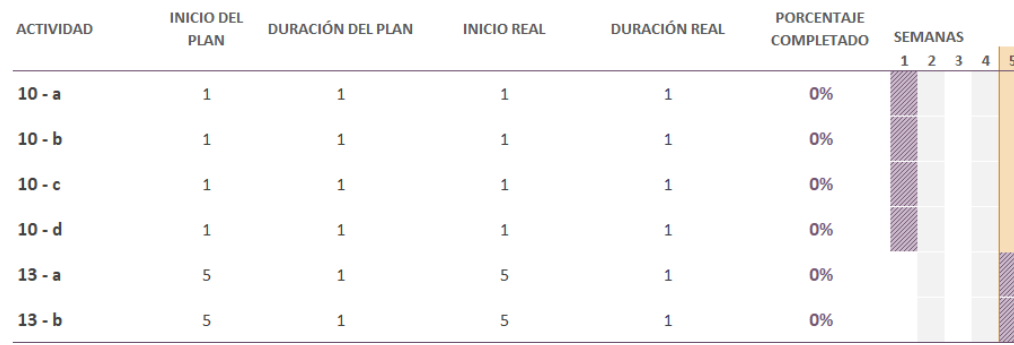


Ilustración 3 Gantt Semanas 1-5 3

Versión Jugable – PEC 3 – 6/04/2020 a 24/05/2020

En esta versión del juego se diseñaron el resto de islas, una especial con recursos no presentes en la isla inicial y otra como isla final del juego. Adicionalmente, se añadieron el resto de acciones del personaje para que el jugador ya tenga todas las capacidades posibles que podrá realizar en la versión final.

Todo el conjunto de elementos vivos, están presentes en esta versión jugable, para que dé una sensación de que el mundo no está vacío, sino que hay elementos vivos con su inteligencia artificial y su comportamiento respecto al jugador.

En esta fase del proyecto, el jugador ya puede realizar construcciones, de forma sencilla para poder realizar acciones, como por ejemplo viajar a otras islas a través de puentes.

Los diferentes menús han sido implementados, para poder tanto pausar el juego como abrir el inventario.

Como últimos pasos previos a la versión final es importante hacer foco en los test para obtener un conjunto de errores encontrados y poderlos corregir con el objetivo de tener un juego lo más pulido posible.

Por último, se realizó el informe correspondiente a esta versión del juego y un vídeo explicativo. Se ha decidido iniciar la memoria dentro de las semanas finales de esta fase, para así adelantarla de cara a la versión final.

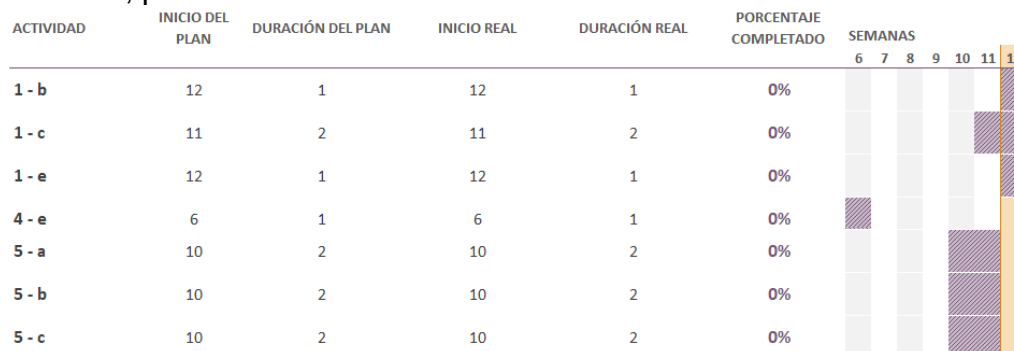


Ilustración 4 Gantt Semanas 6-12 1

ACTIVIDAD	INICIO DEL PLAN	DURACIÓN DEL PLAN	INICIO REAL	DURACIÓN REAL	PORCENTAJE COMPLETADO	SEMANAS							
						6	7	8	9	10	11	12	
5 - d	10	2	10	2	0%								
6 - a	7	2	7	2	0%								
6 - b	7	2	7	2	0%								
6 - c	7	2	7	2	0%								
6 - d	7	2	7	2	0%								
7 - a	7	2	7	2	0%								
7 - b	7	2	7	2	0%								
7 - c	7	2	7	2	0%								
7 - d	7	2	7	2	0%								

Ilustración 5 Gantt Semanas 6-12 2

ACTIVIDAD	INICIO DEL PLAN	DURACIÓN DEL PLAN	INICIO REAL	DURACIÓN REAL	PORCENTAJE COMPLETADO	SEMANAS							
						6	7	8	9	10	11	12	
7 - e	7	2	7	2	0%								
8 - a	9	1	9	1	0%								
8 - b	9	1	9	1	0%								
8 - c	9	1	9	1	0%								
8 - d	9	1	9	1	0%								
8 - e	9	1	9	1	0%								
10 - e	10	2	10	2	0%								
10 - f	6	1	6	1	0%								
10 - g	6	1	6	1	0%								

Ilustración 6 Gantt Semanas 6-12 3

ACTIVIDAD	INICIO DEL PLAN	DURACIÓN DEL PLAN	INICIO REAL	DURACIÓN REAL	PORCENTAJE COMPLETADO	SEMANAS							
						6	7	8	9	10	11	12	
11 - a	11	4	11	4	0%								
11 - b	11	4	11	4	0%								
12 - a	11	4	11	4	0%								
12 - b	11	4	11	4	0%								
13 - c	12	1	12	1	0%								
13 - d	12	1	12	1	0%								
13 - e	11	4	11	4	0%								

Ilustración 7 Gantt Semanas 6-12 4

Versión Final – Entrega final – 25/05/2020 a 07/06/2020

Para esta versión final se ha continuado con los test para poder encontrar más errores y corregirlos.

Por último, se ha continuado con la memoria para poderla terminar a tiempo y con la calidad deseada, adicional al vídeo explicativo y la autoevaluación.

ACTIVIDAD	INICIO DEL PLAN	DURACIÓN DEL PLAN	INICIO REAL	DURACIÓN REAL	PORCENTAJE COMPLETADO	SEMANAS		
						13	14	15
11 - a	11	4	11	4	0%			
11 - b	11	4	11	4	0%			
12 - a	11	4	11	4	0%			
12 - b	11	4	11	4	0%			
13 - e	11	4	11	4	0%			
13 - f	13	1	13	1	0%			
13 - g	14	1	14	1	0%			

Ilustración 8 Gantt Semanas 13-15

1.5 Breve resumen de productos obtenidos

Cada entrega del trabajo de fin de máster ha constado de los siguientes elementos:

- Documento explicativo de los avances.
- Vídeo explicativo de los desarrollos.
- Desarrollo hasta el punto acordado.

En la entrega final del trabajo de fin de máster, se han entregado los siguientes productos:

- Ejecutable del juego BrokenLands, juego de supervivencia en un mundo fragmentado: Se puede encontrar en la carpeta Executable dentro del repositorio del juego.
https://github.com/agonzalezi/TFM_AlexGonzalez/tree/master/Executable
- Enlace y acceso al repositorio GitHub donde está almacenado el código del juego: https://github.com/agonzalezi/TFM_AlexGonzalez
Las clases que conforman el juego han sido comentadas adecuadamente, para mayor entendimiento del juego. La versión final contiene el TAG FinalVersion.
- Memoria del TFM.
- Vídeo presentación del juego y desarrollo: <https://youtu.be/0HUSBkeKSK4>
- Vídeo adicional Full Gameplay: <https://youtu.be/aqQPPk6xwRc>
- Tráiler del juego: <https://youtu.be/hX1kEf60pHQ>
- Autoevaluación.
- Showcase Unity Connect: <https://connect.unity.com/p/brokenland>

1.6 Breve descripción de los otros capítulos de la memoria

- Estado del arte: Un repaso al género del juego, sus influencias y las tecnologías utilizadas en su desarrollo
- Definición del juego: Se explica el origen del juego, su funcionamiento y que puede esperarse el jugador al entrar al mundo de BrokenLand.
- Diseño técnico: Una visión a todas las mecánicas y desarrollos que se han realizado para la construcción del videojuego.

- Diseño de niveles: Visión global de los niveles, para entender, a nivel de diseño, su creación y funcionalidades.
- Manual del usuario: El lector puede encontrar un manual de cómo manejar al personaje en el juego e interactuar con el mundo.
- Conclusiones: Después de la finalización del desarrollo, se ha hecho autocrítica para extraer las conclusiones del trabajo final de máster.
- Glosario: Explicación de la terminología aparecida en la memoria.
- Bibliografía: Recursos de consulta utilizados para la realización del trabajo final de máster.

2. Estado del arte

2.1 Género de juego

BrokenLand se enmarca dentro del género de los denominados *survival*, género popularizado por el conocidísimo Minecraft, que definió y popularizó dicho género; presentando un mundo abierto y lleno de posibilidades al jugador.

Este género se basa en ofrecer al jugador una serie de recursos, que deberá obtener para poder sobrevivir en el mundo que se le presenta. Por lo tanto, no es un género con un objetivo claro, sino que da libertad total al jugador. Si prefiere deambular por el mundo u obtener recursos en un mismo punto, es decisión total del jugador, por lo tanto, es un género muy abierto con el cuál cualquier tipo de jugador puede encontrar su forma de jugar y divertirse.

Por lo tanto, en este género nos encontramos con dos tipos de objetivos:

- **Reto**: Consiste en el objetivo final del juego, algunos *survivals* no lo incluyen, pero sí que es importante plantear al jugador diferentes retos a superar, por ejemplo, llegar a un punto concreto, o matar a un enemigo poderoso (por ejemplo, el dragón de Minecraft) para dar esa sensación de finalización del juego.
- **Supervivencia**: En este género de juego, prima la supervivencia, el jugador deberá buscar recursos para sobrevivir el paso de los días.

Por lo tanto, pese a que el objetivo lo define el jugador en todo momento, sí que el objetivo principal del juego es la supervivencia. El jugador debe buscar formas de poder sobrevivir en el mundo que se le presenta y buscar la forma de satisfacer sus necesidades, siempre simulando las humanas, como, por ejemplo, salud, hambre, temperatura... Uno de los puntos que diferencia los *survivals* en el ámbito de la dificultad es el número de necesidades que hay que satisfacer, debido a que cuantas más necesidades tenga que tener en cuenta el jugador para poder sobrevivir dentro del juego, más difícil será controlarlas; dado que el jugador tendrá que realizar más acciones o buscar más recursos.

BrokenLand ofrece un equilibrio entre estas necesidades (explicadas a continuación) y la libertad al jugador. No es un juego exigente dado que pretende alejarse de otros referentes del género *extreme-survival* como por ejemplo Dont' Starve y ofrecer todo lo contrario. BrokenLand presenta una sensación de supervivencia que no agobia y pretende que el jugador disfrute.

Como referentes dentro del género para el diseño de BrokenLand nos encontramos con juegos indies que han aportado en mayor o menor medida, las ideas base para el desarrollo del juego:

- **Minecraft**: Siempre que pensamos en el juego *survival* por excelencia, viene a la cabeza el juego de Notch. La sensación de libertad que ofrece Minecraft, donde te plantea una serie de mecánicas y es el propio jugador el que determina que hacer a continuación. Si que existe un objetivo global similar a la derrota del dragón en el juego, pero en BrokenLands viene determinada por conseguir abrirse paso a través de las islas hasta llegar a una isla concreta y poder finalizar el juego.



Ilustración 9 Minecraft

- **Ylands**: Este juego desarrollado por los creadores del simulador de guerra Arma, es un ejemplo claro del estilo artístico y las múltiples posibilidades que ofrece BrokenLand. Ylands es un juego que ofrece un estilo artístico LowPoly, que está en auge actualmente en los juegos indie debido a que ofrece modelos 3D, pero si tener gran detalle, ofreciendo un estilo gráfico atractivo, pero sin una gran carga gráfica del juego, permitiendo que sea jugado en múltiples plataformas y sistemas sin tener configuraciones de alta gama. Además, Ylands es un juego de supervivencia, pero en este caso más centrado en la creación de elementos y su distribución con la comunidad; este punto lo dejaremos aparte para un futuro de BrokenLand, debido a que es muy interesante.



Ilustración 10 Ylands

- **Conan Exiles**: Un vasto mundo recreado en 3D donde los jugadores pueden emular las grandes gestas de los cómics de Conan el Bárbaro. Un juego que se ha ido expandiendo a lo largo de los años pero que aún después de varios años de su salida, sigue teniendo una sólida base de jugadores. El sistema de construcción de BrokenLand está claramente inspirado en el sistema de este título, donde los jugadores pueden

construir casas, camas, etc... Un juego exigente dado que requiere de jugar en un servidor compartido multijugador para poder superar los diferentes retos que propone el título.

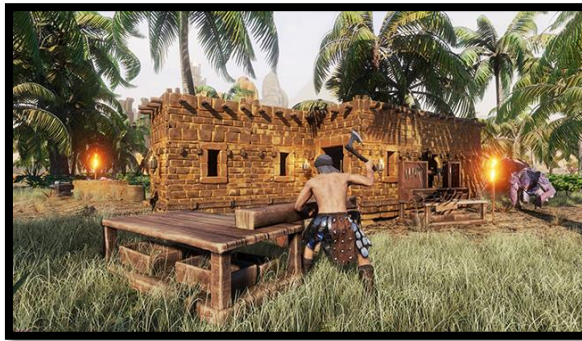


Ilustración 11 Conan Exiles

- **The Long Dark**: Dentro del género de los *survivals* existe una rama donde se encajan aquellos juegos más exigentes para los amantes de la supervivencia extrema. Creado por Hinterland Studio, nos ofrece una experiencia que más se asemeja a imitar la realidad, dado que es un juego que tiene muchas estadísticas que el jugador debe cuidar para poder sobrevivir, como, por ejemplo, hambre, temperatura, etc... Al contrario que BrokenLand, el apartado artístico sería una evolución del LowPoly, debido a que se aleja de modelos realistas con grandes gráficos y se aleja del modelo LowPoly al que estamos acostumbrados en otros juegos, dando un punto de vista diferente a los juegos *survival* y permite ejecutar el juego en configuraciones de baja calidad.



Ilustración 12 The Long Dark

Para poder realizar un correcto análisis de la competencia, primero deberemos listar las características que definen BrokenLand:

- [C1] Supervivencia: La base del juego es la supervivencia del jugador que deberá recolectar comida y evitar a los enemigos para poder sobrevivir en este nuevo mundo.
- [C2] Construcción: El jugador puede usar sus recursos de la forma que guste, como por ejemplo realizar construcciones que le serán útiles para sobrevivir, como por ejemplo una cama.
- [C3] Entorno LowPoly 3D: El mundo presentado al jugador se basa en una estética LowPoly 3D con el objetivo de poder ser ejecutado en configuraciones de baja calidad.
- [C4] Mundo fragmentado: El mundo del juego no está presente de inicio al jugador, sino que es el mismo el que debe ir avanzando para descubrir todos los rincones del mismo.

- [C5] Obtención de recursos: El avatar puede usar sus herramientas para obtener recursos naturales que ofrece el mundo, como por ejemplo piedras o madera.
- [C6] Combate: En el mundo de BrokenLands el combate no es necesario prácticamente, pero sí que ofrece una forma de librarnos de los enemigos que nos atacan.
- [C7] Enemigos pacíficos: Como se ha indicado anteriormente, el objetivo del juego no es hacer sufrir al jugador, por lo tanto, existen animales en los diferentes biomas que no atacarán al jugador.
- [C8] Enemigos agresivos: Pese a que el objetivo del juego no es combatir, el jugador debe tener cuidado con estos enemigos, dado que, si detectan al jugador, le atacarán.
- [C9] Biomas diferenciados: En el mundo fragmentado presente en el juego existen distintos biomas que ofrecen entornos totalmente diferentes, desde islas totalmente desérticas hasta zonas nevadas.

Juego/Característica	C1	C2	C3	C4	C5	C6	C7	C8	C9
Minecraft	X	X			X	X	X	X	X
Ylands	X	X	X		X	X	X	X	X
Conan Exiles	X	X			X	X	X	X	X
The Long Dark	X	X	X		X	X	X	X	

2.2 Tecnologías usadas

Dentro del género de los survival existen múltiples opciones utilizadas por los desarrolladores, basándose más en la experiencia personal de cada uno el elegir uno u otro.

- LWJGL (LightWeight Java Game Library): Para la creación de la primera versión de Minecraft se utilizó esta librería que soporta OpenGL y OpenAL, junto con sistema de input/output para Java. Debido a problemas en el consumo de recursos, se desarrolló a posterior una nueva versión para Windows 10 basada en C++.
- Unity: Utilizado por un amplio número de juegos indie, nos encontramos que Ylands y The Long Dark lo usaron en su desarrollo. Debido a la gran comunidad y las facilidades que ofrece el engine, muchos desarrolladores indies se decantan por él.
- Unreal Engine: De todos los juegos revisados en el estudio del arte, el juego de Conan Exiles es el que usa este motor. Unreal ofrece un motor robusto, con una gran potencia gráfica y la posibilidad de programar de forma gráfica basándose en nodos, ideal para una persona no versada en el mundo de la programación.

3. Definición del juego

3.1 Descripción

BrokenLand es un juego de supervivencia en el que el jugador deberá buscar recursos que le permitan sobrevivir y descubrir los misterios que oculta este

nuevo mundo al que se tiene que enfrentar. La supervivencia del jugador dependerá tanto de los recursos que pueda conseguir, como de la unificación del mundo del juego; fragmentado después de un cataclismo desconocido ocurrido después de un gran flash. Este fenómeno alteró la estructura del mundo para siempre...

3.2 Tipo de interacción juego-jugador

El jugador controla a un avatar representado por un humano que puede moverse libremente por el mundo.

Para poder sobrevivir, el jugador debe buscar recursos y para ello puede realizar diferentes acciones que le permitan obtenerlos. Las formas en las que se pueden obtener recursos son:

- Acción de recoger: Coger un objeto del suelo.
- Acción de talar: Si el jugador consigue hacerse con un hacha, puede talar los árboles.
- Acción de picar: Si el jugador consigue hacerse con un pico, puede picar piedra.

Posteriormente, un objeto puede ser usado, por ejemplo, para comerlo.

Más allá de la recolección de recursos, el jugador puede realizar construcciones si dispone de los materiales adecuados.

Uno de los objetivos principales del juego es la unificación del mundo y llegar a una isla final. Para poder unir las distintas islas, y poder obtener nuevos recursos se deberán construir puentes en puntos determinados. Una vez el puente esté construido el jugador puede viajar libremente a la siguiente isla.

Es un juego de mundo abierto, por lo tanto, el jugador puede en todo momento viajar por las distintas islas.

El juego tiene una interfaz a modo de inventario de lo que tiene el jugador disponible para usar. A modo de HUD, se muestran las estadísticas del avatar por pantalla, para que el jugador tenga mayor control para poder cubrir sus necesidades.

3.3 Plataforma de destino

Inicialmente se va a desarrollar una versión para un PC Windows de sobremesa. Pero debido a que va a ser un juego con una baja carga gráfica, no se descarta evolucionarlo en un futuro para desarrollar una versión de móvil.

3.4 Historia y ambientación

El mundo colapsó de un momento a otro, no se sabe muy bien porque, un fogonazo en el cielo y todo cambió; el mundo tal y como lo conocíamos ya no existía y la humanidad tuvo que adaptarse a este nuevo mundo. El mundo ha vuelto a su estado original, como si se hubiera hecho un reinicio de todo, no existe la tecnología y la humanidad debe adaptarse para poder sobrevivir.

Más allá del mundo, la humanidad ha perdido todos los recuerdos anteriores al colapso, nada de su vida pasada ni de sus vivencias, es más, todo ha sido borrado de la faz de la tierra, edificios, coches, etc... La humanidad deberá volver a aprender de nuevo, sin saber a ciencia cierta si el futuro que tienen por delante es próspero o presentará múltiples dificultades.

El protagonista de nuestra historia, se despierta justo en el momento después del colapso, sin recordar nada y deberá volver a empezar, buscar la forma de sobrevivir en este nuevo mundo. Pero más allá de la supervivencia, el protagonista se pregunta ¿Qué habrá en esas misteriosas islas que se ven en el horizonte?

3.5 Definición de los personajes y elementos

En BrokenLand, el personaje principal es el único humano que aparece en el juego.

Pero no es el único personaje vivo, sino que se encontrará con animales y quien sabe qué, más allá de la isla inicial. Con estos elementos vivos del juego, el jugador podrá interactuar y acabar con ellos para poder conseguir recursos.

Además de los elementos vivos, existen otros elementos en el mundo, con los cuales el jugador puede interactuar:

- Árboles: Es posible obtener comida, que cae de los mismos, o talarlos para poder obtener madera.
- Menas: Pequeños montículos en los que conseguir piedra.
- Arbustos: Uno de los elementos esenciales, al menos al inicio del juego, dado que producen la primera fuente de comida que descubre el jugador, las bayas.
- Construcciones: Una vez construimos algún mueble, como mínimo una cama, podremos interactuar con ellos para realizar algunas acciones, como dormir.
- Puentes: El jugador puede construir un puente que conectará dos islas y de esta forma viajar entre ellas.
- Comida: Uno de los elementos más importantes del juego, el jugador puede interactuar con la comida, para comerla y así reducir el hambre.

3.6 Interacción entre los actores del juego

Como actores dentro de BrokenLand, tenemos tres bien diferenciados:

- Avatar: Es el modelo del personaje que el jugador usa para jugar, puede moverse y realizar diversas acciones, todo para poder sobrevivir en este mundo.
- Elementos vivos pacíficos: Son elementos que deambulan por la isla, pero que no atacan al jugador, debido a porque no son agresivos o porque simplemente son herbívoros y no necesitan atacar al jugador.
- Elementos vivos agresivos: Son elementos, por ejemplo, animales, que al ver al jugador se lanzan a por él para matarlo. Hay que tener muchísimo cuidado dado que enfrentarse a estos enemigos es difícil dado que como se ha explicado anteriormente, los recursos son escasos y el jugador deberá determinar si merece la pena o no, enfrentarse a estos enemigos para conseguir más recursos o es mejor huir.

Adicionalmente tenemos otros elementos dentro del juego:

- Elementos de los que obtener recursos: Árboles, menas o arbustos.
- Elementos para recoger: El jugador puede obtener estos objetos del suelo, como por ejemplo las manzanas.

- Construcciones: Con todas las construcciones, el jugador puede interactuar. De esta forma se da un uso real a cada construcción.

3.7 Objetivos planteados al jugador

Uno de los conceptos clave del género *survival* es el de que sea el propio jugador el que plantee sus propios objetivos, pero para este trabajo de final de máster, se ha diseñado un objetivo principal para poder finalizar el juego que cumplimenta al objetivo inherente del género en sí:

- Sobrevivir: El jugador deberá preocuparse de sus estadísticas para poder sobrevivir. En una primera versión del juego, nos encontramos con:
 - o Salud: Deberá preocuparse de no perderla o el avatar muere.
 - o Hambre: El jugador debe recoger recursos para alimentarse, una vez está lleno el indicador de hambre, el jugador comienza a recuperar salud.

Si el jugador pierde toda su salud, pierde la partida, teniendo que volver a comenzar.

- Isla destino: Una isla en la cual avanzará la historia dentro del juego debido a que se descubre que algo raro pasa en el mundo del juego, debido a unas extrañas estructuras que no parecen de este tiempo..., pero en la cual la versión actual finalizará. El objetivo de esta versión del juego, es llegar a la isla de destino teniendo que sobrevivir los suficientes días como para poder llegar a ella.

3.8 Concept Art

El estilo artístico es LowPoly en 3D, como, por ejemplo, los paquetes Polygon, de la empresa Synty Studios que están en la Asset Store.



Ilustración 13 Estilo LowPoly de los paquetes Polygon

La idea es que cada isla sea única, y que sean flotantes, como la siguiente imagen.



Ilustración 14 Ejemplo de isla flotante

Las islas serán como un mundo fragmentado, como en la siguiente imagen de las Llanuras Quebradas de la saga del Archivo de las Tormentas; pero con más separación.



Ilustración 15 Llanuras Quebradas

4. Diseño técnico

4.1 Entorno

A lo largo del primer semestre del máster, se utilizó el motor Unity en varias de sus versiones, aprendiendo desde lo más básico de un juego en dos dimensiones a juegos complejos como por ejemplo un juego online. Adicionalmente al conocimiento previo, una de las cosas que me atrae más de Unity es la de la enorme comunidad que hay detrás y la enorme biblioteca de información que es Unity Learn por ejemplo, que incluso tiene infinidad de seminarios, gamejams, etc...

Unity también ofrece multitud de contenido gratuito (modelos, animaciones, efectos, etc...) para realizar pruebas o desarrollar pequeños proyectos, dentro de la Asset Store que van a ser de gran apoyo para no tener que buscar a lo largo de Internet por Assets que encajen bien con el juego a implementar, sino que en un mismo lugar accesible desde Unity puedes buscarlos y testarlos.

Al inicio del proyecto la versión más actualizada de Unity era la versión del motor 2019.3.2f1, de esta forma se tendrán a disposición todos los nuevos agregados de la versión 2019 del engine. Se ha continuado durante todo el desarrollo con esta misma versión para evitar problemas de incompatibilidad con las mecánicas desarrolladas y los Assets utilizados.

Para la edición de código se ha usado el IDE que viene integrado con la instalación de Unity, Visual Studio 2019.

4.2 Requerimientos técnicos del entorno

Pese a que Unity está preparado para un consumo de recursos bajo, se ha utilizado una configuración de alta gama para el desarrollo, para evitar tener problemas a la hora de renderizar y jugar al juego. No se ha usado ningún hardware adicional, más allá de un ordenador. Los requerimientos que recomienda el entorno utilizado y los que se han usado son:

Requerimiento	Mínimo recomendado	Utilizado
Sistema Operativo	Windows 7/10 de 64 bits	Windows 10 de 64 bits
CPU	Arquitectura de x64	Arquitectura de x64
API Gráfica	DirectX 10, 11 o 12	DirectX 12

4.3 Herramientas empleadas

Para el desarrollo del proyecto se han utilizado las siguientes herramientas:

- LunaPic (<https://www.lunapic.com/editor/>): Este potente editor te permite sin tener que realizar la descarga de ningún software, editar imágenes e incluso realizar un tratamiento avanzado de las mismas, como, por ejemplo, eliminar el fondo de una imagen para hacerlo transparente.
- Paint3D: Herramienta integrada en Windows 10 que permite rediseñar los modelos 3D utilizados en el juego, ya sea para cambiar algún color o realizar algún retoque.
- Canva (<https://www.canva.com/>): Un editor de imágenes online que es ofrecido por la versión de estudiantes de Unity y permite realizar montones de composiciones con distintos elementos. Desde imágenes superpuestas a la posibilidad de encuadrar una imagen.
- Visual Studio 2019: El IDE de Microsoft que está integrado en la instalación de Unity ha sido el utilizado para el desarrollo del código del juego.
- LunaPic (<https://www.lunapic.com/editor/>): Este potente editor te permite sin tener que realizar la descarga de ningún software, editar imágenes e incluso realizar un tratamiento avanzado de las mismas, como, por ejemplo, eliminar el fondo de una imagen para hacerlo transparente.
- Unity 3D: El editor usado explicado en el apartado [4.1](#) para el desarrollo del juego en su totalidad, en su versión 2019.3.2f1.
- GitHub: El repositorio del juego está creado dentro de GitHub, esta herramienta me ha permitido, no solo almacenar el código, sino organizar las distintas *releases* del juego de cara a tener una mayor organización en el mismo.
- GitKraken: Para la sincronización del código se ha utilizado esta herramienta que ya se usó en el primer semestre de este curso y ha permitido la subida de las diferentes versiones del juego, al repositorio de GitHub.

4.4 Recursos utilizados

La mayoría de los recursos del juego han sido obtenidos de forma gratuita, pero debido al momento que nos encontramos, hay empresas que han ofrecido sus Assets de forma gratuita temporalmente (Kenney) o de pago para recaudar dinero para una asociación benéfica (Synty Studios). De esta manera, los recursos han sido obtenidos de las siguientes fuentes:

- Kenney (<https://www.kenney.nl/>): Una empresa desarrolladora de Assets que tiene tienda propia en su web. Se encarga mayormente de ofrecer multitud de recursos gratuitamente, pese a que hay algunos packs grandes que son de pago. Este año ha cumplido 10 años y lo ha querido celebrar ofreciendo sus packs de pago, sin coste (<https://itch.io/s/29028/kenney-10-year-anniversary>).
- OpenGameArt (<https://opengameart.org/>): La comunidad de Assets de videojuegos gratuitos por excelencia, una de las grandes fuentes de

recursos que me han ayudado a lo largo del curso; donde podemos encontrar desde audio, diseños 2D y 3D, e incluso texturas.

- Asset Store (<https://assetstore.unity.com/>): La tienda de Assets integrada en Unity no ha sido muy útil dentro de este proyecto, debido al poco material gratuito que encajara con el proyecto dentro de la misma.
- Quaternius (<http://quaternius.com/>): Un diseñador gráfico dedicado casi en exclusiva al mundo LowPoly, ofrece multitud de Assets de forma gratuita para enriquecer la comunidad LowPoly dentro de los desarrolladores de juegos.
- Mixamo (<https://www.mixamo.com/>): Esta herramienta online ofrecida por Adobe permite obtener de forma gratuita multitud de animaciones para nuestro personaje del juego.
- Synty Store (<https://syntystore.com/>): Tienda propia de una empresa de diseño gráfico especializada en un estilo LowPoly 3D. Junto con Humble Bundle, ofrecieron mucho material a cambio de una aportación para una asociación benéfica (<https://www.humblebundle.com/software/best-of-polygon-game-dev>).
- Google Poly (<https://poly.google.com/>): Web ofrecida por Google para que diseñadores gráficos muestren sus obras, en su gran mayoría usando el estilo LowPoly.
- Youtube (<https://www.youtube.com/>): Dentro de Youtube, hay una gran biblioteca de audio libre para el uso de los creadores de contenido.
- 1001 Fonts (<https://www.youtube.com/>): Gran catálogo de fuentes.

Los recursos utilizados para el desarrollo de BrokenLands se describen a continuación:

1. Polygon Adventure Pack
 - a. Fuente: Synty Store
 - b. Link: <https://syntystore.com/products/polygon-adventure-pack>
 - c. Descripción: Pack de modelos LowPoly 3D ambientados en juegos de aventuras.
 - d. Recursos utilizados: Se han utilizado los modelos de los entornos de las primeras dos islas, el modelado del personaje y algunos objetos.



Ilustración 16 Polygon Adventure Pack

2. Polygon Vikings Pack
 - a. Fuente: Synty Store
 - b. Link: <https://syntystore.com/products/polygon-vikings-pack>

- c. Descripción: Pack de modelos LowPoly 3D ambientados en la era vikinga.
- d. Recursos utilizados: Se han utilizado los modelos de los entornos de la última isla, y las construcciones.



Ilustración 17 Polygon Vikings Pack

- 3. Polygon Prototype Pack
 - a. Fuente: Synty Store
 - b. Link: <https://syntystore.com/products/polygon-prototype-pack>
 - c. Descripción: Pack de modelos LowPoly 3D para el prototipado.
 - d. Recursos utilizados: Algunos iconos y letras han sido utilizados para los diferentes menús del juego.



Ilustración 18 Polygon Prototype Pack

- 4. Cartoon FX Free
 - a. Fuente: Asset Store
 - b. Link: <https://assetstore.unity.com/packages/vfx/particles/cartoon-fx-free-109565>
 - c. Descripción: Pack de efectos, con aspecto de dibujos animados.
 - d. Recursos utilizados: Se han utilizado algunos efectos del pack para los efectos del juego como, por ejemplo, al caer al vacío.



Ilustración 19 Cartoon FX Free

- 5. Farm Animal Pack
 - a. Fuente: Quaternius
 - b. Link: https://drive.google.com/drive/folders/1is1ax2V0mIEDS7BbbzBV_x2Pabya1dVJ
 - c. Descripción: Pack de animales de granja con estilo LowPoly.

- d. Recursos utilizados: Se han usado tanto los modelos como algunas animaciones del caballo y la llama.



Ilustración 20 Farm Animal Pack

6. Easy Enemy Pack

- a. Fuente: Quaternius
b. Link: <https://drive.google.com/drive/folders/1VbJlslXPWK-1KybQN6yezZrfJcw608qe>
c. Descripción: Pack de enemigos de tamaño pequeño.
d. Recursos utilizados: Se han usado tanto los modelos como algunas animaciones de los animales agresivos, la araña y la serpiente.



Ilustración 21 Easy Enemy Pack

7. Food Pack vol. 1

- a. Fuente: Quaternius
b. Link: http://www.mediafire.com/file/y661nazt4o5k127/Food_Pack_Vol.1.rar/file
c. Descripción: Modelos de comida con estilo LowPoly.
d. Recursos utilizados: El modelo del muslo de carne, indicador del hambre en el juego, está extraído de este pack de Assets.

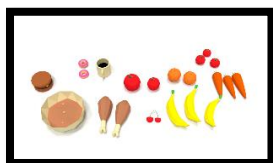


Ilustración 22 Food Pack

8. PowerUps

- a. Fuente: Quaternius
b. Link: http://www.mediafire.com/file/m44ksskxk8887lh/PowerUps_by_Quaternius.rar/file
c. Descripción: Modelos de iconos de PowerUp con estilo LowPoly.
d. Recursos utilizados: El modelo del corazón, indicador de la vida del protagonista del juego, está extraído de este pack de Assets.

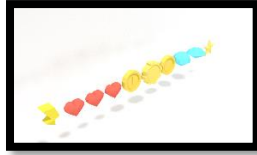


Ilustración 23 PowerUps

9. Space Pack vol. 1
 - a. Fuente: Quaternius
 - b. Link: <http://www.mediafire.com/file/617128jee6d7mg1/SpacePack.rar/file>
 - c. Descripción: Modelos 3D con estilo LowPoly de temática espacial.
 - d. Recursos utilizados: El modelo del planeta tierra de la introducción está extraído de este pack.

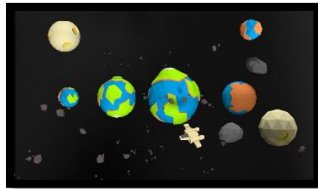


Ilustración 24 Space Pack

10. Third Person Controller - Basic Locomotion FREE
 - a. Fuente: Asset Store
 - b. Link: <https://assetstore.unity.com/packages/tools/utilities/third-person-controller-basic-locomotion-free-82048>
 - c. Descripción: Versión gratuita de un avatar en tercera persona.
 - d. Recursos utilizados: Este pack es el recomendado por Unity, que vendría a substituir al StandardAssets utilizado en el curso, pero para la versión 2019. Se han utilizado los controladores de la cámara y el personaje, juntamente con las animaciones básicas de movimiento. Es un pack limitado, por lo tanto, se debe tomar como template para adaptarlo a las necesidades de cada uno.



Ilustración 25 Basic Locomotion Free

11. Modelos del sol, la luna y las estrellas
 - a. Fuente: Google Poly
 - b. Link: <https://poly.google.com/view/69ejysWdDXG>
<https://poly.google.com/view/9OPocAqXM0u>
<https://poly.google.com/view/5hrwvzpCArU>
 - c. Descripción: Modelos 3D con estilo LowPoly.
 - d. Recursos utilizados: Son los iconos que aparecen para indicar el momento del día y las estrellas que aparecen en la introducción.



Ilustración 26 Luna



Ilustración 27 Sol



Ilustración 28 Estrella

12. Kenney Game Assets

- Fuente: Kenney
- Link: <https://kenney.itch.io/kenney-game-assets-1>
- Descripción: Pack de assets variados.
- Recursos utilizados: Se han utilizado los efectos de sonido del pack para representar las acciones del jugador y dar mayor inmersión.

13. Rabbit Eating

- Fuente: OpenGameArt
- Link: <https://opengameart.org/content/rabbit-eating>
- Descripción: Sonido de un conejo comiendo.
- Recursos utilizados: Utilizado para la acción de comer del personaje.

14. Nihilore – The Age of Wonder

- Fuente: OpenGameArt
- Link: <https://opengameart.org/content/the-nihilore-collection>
- Descripción: Música de corte medieval.
- Recursos utilizados: Es el tema principal del juego que puede escucharse en el menú del mismo.

15. Memoraphile – Up in the Sky

- Fuente: OpenGameArt
- Link: <https://opengameart.org/content/up-in-the-sky>
- Descripción: Música ambiental.
- Recursos utilizados: Es el tema que suena durante el transcurso del juego para dar más inmersión al jugador.

16. Jeremy Blake – Up in the Sky

- Fuente: Youtube
- Link: <https://www.youtube.com/audiolibrary/>
- Descripción: Música espacial con un toque de misterio.

- d. Recursos utilizados: Ha sido usado para ambientar el vídeo introductorio del juego.

17. Broken Glass

- a. Fuente: 1001 Fonts
- b. Link: <https://www.1001fonts.com/broken-glass-font.html>
- c. Descripción: Fuente de texto con aspecto de cristal roto.
- d. Recursos utilizados: El título de BrokenLands está creado con esta fuente para dar la sensación de fragmentación.



Ilustración 29 BrokenGlass Font

18. Underdog

- a. Fuente: 1001 Fonts
- b. Link: <https://www.1001fonts.com/underdog-font.html>
- c. Descripción: Fuente de texto con aspecto de dibujos animados.
- d. Recursos utilizados: Encajaba muy bien con el estilo del juego así que se decidió usar esta fuente adicional, para que el juego tuviera un tono desenfadado.



Ilustración 30 Underdog Font

19. Animaciones de las acciones del personaje

- e. Fuente: Mixamo
- f. Link: <https://www.mixamo.com/>
- g. Descripción: Multitud de animaciones para personajes.
- h. Recursos utilizados: Las animaciones para las acciones del personaje se han extraído de la página de Adobe Mixamo.

La mayoría de los Assets utilizados estaban desfasados (se entiende que se crearon hace tiempo) y no funcionaban correctamente con la versión de Unity actual, por ello se han tenido que modificar para su uso:

- Animaciones de los animales: La mayoría de animaciones no funcionaban de forma correcta, debido a que habían frames mal montados y creaban un comportamiento errático e incluso sin sentido al realizar la animación de forma continua en loop. Se han rediseñado para adecuarlas al comportamiento esperado, adicionalmente se han completado para realizar una animación continua y que no se detecten cortes en la animación.
- Adaptación de animaciones: Existen algunas animaciones de los animales que no existían en los packs de Assets usados, como por ejemplo el movimiento de caminar de las llamas. Para ello se han tomado

unas animaciones de otro animal y se han adaptado para aquellos animales que no tenían animación. Dicha adaptación se ha hecho frame a frame, para revisar que cada punto del objeto funcionara correctamente dentro de la animación.

- Adaptación y mejora del manager de la cámara: El Asset de la cámara en tercera persona usado, presentaba numerosos problemas, por ejemplo, al chocar con objetos o hacer movimientos inesperados al cruzar un puente. Por ello se ha tenido que revisar el código y adaptarlo al objetivo deseado, para pulir estos detalles.
- Adaptación del controlador del movimiento del personaje: Se ha tenido que modificar el controlador del movimiento del personaje, para adaptarlo al funcionamiento del juego en diferentes situaciones, por ejemplo, al saltar, o al abrir algún menú.

Algunos recursos se han creado de 0 debido a que no se encontraba algo que se adaptara a las necesidades del juego, en tiempo de desarrollo:

- Animaciones de la caída de los diferentes árboles del juego.
- Animaciones de muerte de algunos animales, como por ejemplo la llama, no existente en el pack de Assets usado.
- Iconos y menús, creados con las herramientas online mencionadas en el apartado [4.3](#).
- Sistema de Fade-In, Fade-out; existían Assets para realizar estas transiciones, pero para tener mayor control sobre lo que se realizaba, se ha decidido realizarlo desde 0 con código y animaciones.
- Modelos de arbusto de bayas, árbol frutal y menas de piedras: Unificando diferentes modelos se han creado los diferentes modelos que no estaban presentes en los packs de Assets anteriormente mencionados.

4.5 Escenas/Pantallas

Menú principal: Permite acceder al menú de opciones que permite configurar tanto la dificultad como la necesidad de mostrar las guías del juego o saltar la introducción; acceder a la pantalla de controles para aprender a jugar, iniciar un nuevo juego y salir del juego. Se ha diseñado mediante iconos, para que la interfaz sea lo más intuitiva posible. El fondo de los menús, viene a representar el vacío que es lo que se encuentra al jugador, dentro del mundo que plantea el juego.



Ilustración 31 Menú Principal

Los dos iconos inferiores, abren y cierran, los menús de configuración y la visualización de los controles.

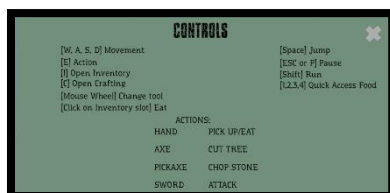


Ilustración 32 Controles



Ilustración 33 Opciones

Introducción: Breve animación para introducir al mundo del juego, al jugador. De esta forma se consigue que el jugador sepa que se va a enfrentar a un mundo totalmente diferente al que está acostumbrado.



Ilustración 34 Introducción

Escena del juego: Es la pantalla principal del juego, donde podemos mover al jugador, ver el HUD y caminar por el mundo. Permite girar la cámara con el ratón.



Ilustración 35 Escena de juego

Pulsando la tecla ESC o la letra P, podemos abrir y cerrar el menú de pausa.



Ilustración 36 Menú de pausa

Pantalla final: El juego puede finalizar por dos motivos, el jugador pierde su vida y muere, por lo tanto, en esta pantalla aparecería el mensaje de "GAME OVER",

pero si el jugador ha llegado a la brecha temporal, aparecerá un mensaje como se puede ver en la imagen.



Ilustración 37 Pantalla final

4.6 Diagrama de componentes

En el diagrama a continuación, podemos ver a varios niveles, los diferentes componentes que componen BrokenLand. Los componentes han sido divididos por niveles para resaltar su implicación transversal en otros componentes. Los niveles definidos son:

- Nivel 1: Aquellos componentes que afectan a todo conjunto global del juego.
- Nivel 2: Los componentes que controla o ve el jugador, de forma directa.
- Nivel 3: Componentes que están estrechamente relacionados con el nivel superior, ya sea porque son contenidos o derivados de otro componente.

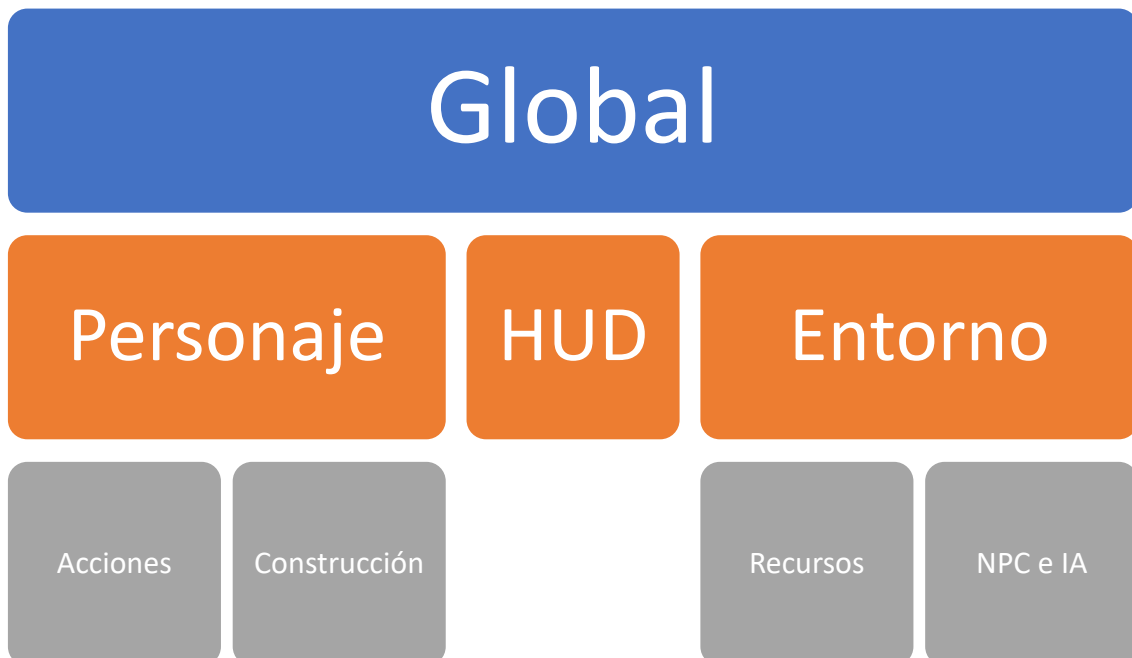


Ilustración 38 Diagrama de componentes

Los componentes que componen BrokenLand son:

- Global: Contiene aquellas mecánicas globales que afectan a todo el juego, como es el control del mismo, o los menús.
- Personaje: Todos aquellos desarrollos que engloban al manejo del personaje por parte del jugador y su interacción.

- Acciones: Las acciones que el jugador puede realizar vienen incluidas en este componente.
- Construcción: Más allá de sobrevivir, el jugador puede realizar construcciones que se detallan en este componente.
- HUD: Los indicadores que ve el jugador como feedback del juego, vienen incluidos en este componente, como, por ejemplo, la vida o el hambre.
- Entorno: El jugador puede moverse por el entorno que ofrece el juego, de tres islas conectadas por puentes.
 - Recursos: En las diferentes islas del juego, el jugador puede encontrarse con recursos que le ayudarán a sobrevivir.
 - NPC e IA: El jugador puede encontrarse, durante su aventura, a varios animales controlados que poseen una IA propia.

4.7 Global

El componente global afecta a todo el conjunto del juego y permiten que se desarrolle el mismo. A continuación, se explican los subcomponentes que lo conforman:

Opciones:

Desde el menú principal, el jugador puede acceder al menú de opciones del juego, para realizar diferentes ajustes que le permitirán cambiar parámetros del juego:

- Dificultad: Existen tres rangos de dificultad dentro del juego para que cada jugador lo configure a su gusto, de esta forma se consigue que el jugador pueda jugar a su ritmo, sin tener que enfrentarse a un reto demasiado difícil.

Existen tres rangos de dificultad:

1. Easy:
 - a. Cada 3 minutos se pierde un punto de la estadística del jugador, dependiendo de si tiene hambre (perderá puntos de vida) o no tiene (perderá puntos de hambre).
 - b. La regeneración de los árboles y las menas de piedra se realizarán pasados 1 minuto.
 - c. La regeneración de las manzanas, de los árboles frutales, se realizará cada 2 minutos.
2. Normal:
 - a. Cada 1 minutos se pierde un punto de la estadística del jugador, dependiendo de si tiene hambre (perderá puntos de vida) o no tiene (perderá puntos de hambre).
 - b. La regeneración de los árboles y las menas de piedra se realizarán pasados 5 minutos.
 - c. La regeneración de las manzanas, de los árboles frutales, se realizará cada 10 minutos.
3. Hard:
 - a. Cada 30 segundos se pierde un punto de la estadística del jugador, dependiendo de si tiene hambre (perderá puntos de vida) o no tiene (perderá puntos de hambre).

- b. La regeneración de los árboles y las menas de piedra se realizarán pasados 10 minutos.
 - c. La regeneración de las manzanas, de los árboles frutales, se realizará cada 20 minutos.
- Omisión de la introducción: Permite al jugador seleccionar si quiere ver el vídeo de introducción o quiere saltárselo para pasar directamente al juego.
 - Inclusión de guías: El jugador puede seleccionar si tener una guía durante todo el juego (no es obligatorio seguirlas) para que le facilite el saber que hacer a continuación para finalizar el juego.

Guías del juego:

Para guiar al jugador durante su aventura y evitar esa sensación de pérdida de objetivo, se han incorporado cuatro guías al juego. Estas guías no son obligatorias para que el jugador avance en su aventura, pero sí que ayudan al tipo de jugador que necesita estímulos, como por ejemplo objetivos, a estar pegado al juego. Aparecen en la parte superior derecha de la pantalla de juego de forma permanente, para que el jugador no tenga que preocuparse de abrir un menú secundario y perder el foco en el juego.

Las diferentes guías engloban cada una diferentes tareas, una vez el jugador realice todas las tareas de una guía, se le mostrará como completa y le aparecerá la siguiente.

Las guías abarcan desde acciones simples para que el jugador se familiarice con el sistema de juego, como talar un árbol, a la finalización del juego, llegando a la brecha.

Estas guías se describen a continuación:

- Guía 1: Son los primeros pasos que debe realizar el jugador para poder sobrevivir en el mundo del juego.
 - Encontrar un hacha.
 - Encontrar un cuchillo.
 - Encontrar un pico.

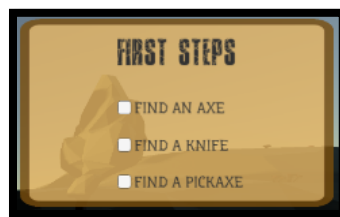


Ilustración 39 Guía 1

- Guía 2: Una vez el jugador ha conseguido encontrar las herramientas básicas, deberá aprender a obtener recursos con las herramientas recién conseguidas.
 - Coger una baya.
 - Romper una roca con el pico.
 - Talar un árbol.



Ilustración 40 Guía 2

- Guía 3: El objetivo de esta guía es explicarle los diferentes menús al jugador, para que entienda las opciones que tiene y que no están a simple vista, como el inventario o el menú de crafting.
 - Abrir el menú de crafting con la tecla C.
 - Abrir el menú de inventario con la tecla I.
 - Aprender a comer a través del inventario, haciendo click en la baya que hay en el inventario, o mediante los accesos rápidos.



Ilustración 41 Guía 3

- Guía 4: Una vez el jugador ya ha aprendido las mecánicas básicas del juego, la última guía invita a la exploración y a cómo terminar el juego.
 - Construir el primer puente entre la isla 1 y 2.
 - Construir el primer puente entre la isla 2 y 3.
 - Encontrar la brecha que hará que el jugador gane la partida y finalice el juego.



Ilustración 42 Guía 4

Sistema de supervivencia:

Como en todos los juegos del género *survival* el personaje controlado por el jugador va perdiendo algunas estadísticas, con tal de obligar al jugador a poner remedio a la situación buscando recursos que le ayuden a sobrevivir.

En BrokenLand existen dos estadísticas que simulan la sensación de supervivencia dentro del juego:

- Vida: Representado por corazones y con un máximo de 5, indican el nivel de vida que tiene el jugador. Es la estadística más importante, debido a que, si pierde todos los corazones, el jugador muere y debe volver a empezar.

- **Hambre:** Representado por unos iconos de muslos de pollo y con un máximo de 5, informan del nivel de hambre de jugador. Por lo tanto, cuán satisfecho está, por lo tanto, si llega a 0, el jugador estará hambriento.

Derivados de estas estadísticas existen dos estados en los que el jugador puede estar:

- **No hambriento:** El jugador tiene sus niveles de hambre entre 1 y 5. Perderá el nivel de hambre cada poco tiempo.



Ilustración 43 Estado no hambriento

- **Hambriento:** Con todos los niveles de hambre consumidos, el jugador pasa al estado de hambriento y empieza a perder vida cada poco tiempo.



Ilustración 44 Estado hambriento

Se ha diseñado un sistema basado en una unidad de tiempo (diferente en cada nivel de dificultad), que hará que el jugador vaya perdiendo una unidad de cada estadística (en base a su estado).

El ciclo de supervivencia del jugador viene descrito en el siguiente diagrama:



Ilustración 45 Ciclo de supervivencia

Por lo tanto, es un factor esencial el que el jugador coma los diferentes recursos disponibles de comida para evitar que su nivel de hambre baje a 0. En caso de que el jugador esté en estado hambriento y coma algo, el jugador pasará al estado de no hambriento y ya no perderá vida en cada unidad de tiempo, sino que perderá una unidad de hambre.

El ciclo de lo que ocurre en cada unidad de tiempo (de pérdida de estadística) es el siguiente:

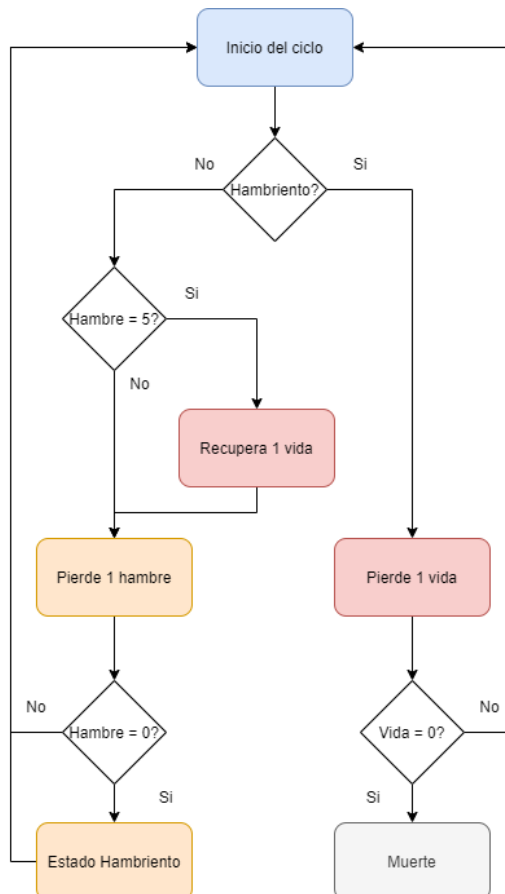


Ilustración 46 Diagrama de supervivencia

Como se puede ver en el diagrama superior, existe una forma de recuperar vida dentro del juego y es mantenerse siempre con el nivel de hambre al máximo, de esta forma, en cada ciclo, el jugador recuperará un punto de vida.

Ciclo día y noche:

Para dotar de más realismo al mundo de BrokenLand se ha creado un sistema de día y noche, que se informa en el HUD del jugador. Adicionalmente a los cambios estéticos del ciclo día y noche, en los elementos del juego, la idea detrás de la implementación del sistema de día y noche es la de mostrar al jugador que el tiempo pasa y al pasar el tiempo, sus estadísticas bajan; como en la vida real, si pasa mucho tiempo, las personas pasamos hambre.

Al inicio del desarrollo no se planificó, pero después de muchos test, se ha detectado que si no hay ningún elemento que simule el paso del tiempo, el jugador parece que pierde /hambre vida sin motivo. La inclusión del elemento de tiempo al juego ha sido necesaria, para que el jugador tenga una referencia del tiempo que ha pasado.

Este ciclo se ha conseguido con tres elementos:

- Objeto sol: Una luz con luz amarilla de alta intensidad.
- Objeto luna: Una luz azul de baja intensidad que simula la luz de la luna, para evitar que el juego se quede totalmente a oscuras por la noche. Siempre enfoca a las islas para dar la luz, durante toda la noche.



Ilustración 47 Objetos sol y luna

- Función de rotación: Para simular el día y la noche dentro del juego, se ha ido rotando el objeto sol sobre el eje X, cada poco tiempo. Adicionalmente a la rotación del objeto, se ha cambiado su intensidad; por lo tanto, el objeto sol tendrá dos estados:
 - Día: Intensidad máxima, rotación entre 0 y 180 grados.
 - Noche: Intensidad mínima, rotación entre 180 y 360 grados.

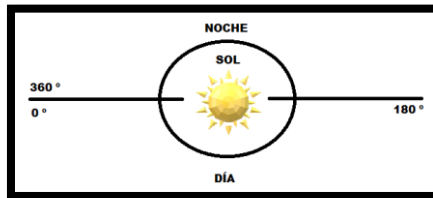


Ilustración 48 Diagrama de rotación del sol

Con la función de rotación conseguimos que visualmente se vean los cuatros estados del día en el juego:

Amanecer y día:

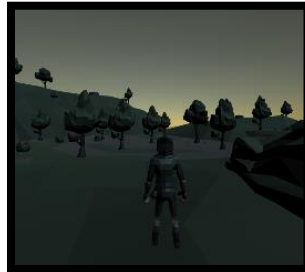


Ilustración 49 Amanecer

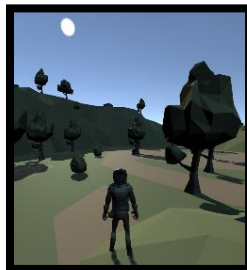


Ilustración 50 Día

Anochecer y noche:

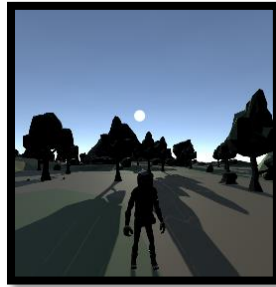


Ilustración 51 Anochecer

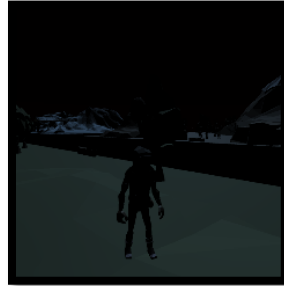


Ilustración 52 Noche

Finalización del juego:

En el juego, existen diferentes formas de finalizar el juego. El jugador debe buscar en todo momento el objetivo del juego, que es llegar a la brecha que se esconde en la tercera isla.

El jugador deberá saltar por las distintas plataformas para llegar a la cima de la estructura de piedra, donde se encontrará con un campamento vikingo derruido y la brecha temporal, que indica el final del juego en la versión actual.

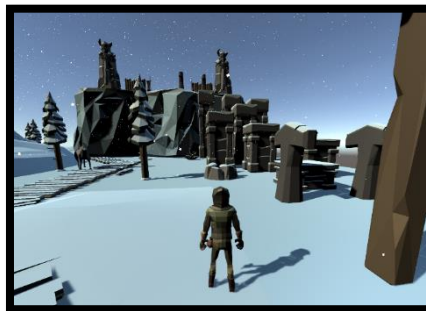


Ilustración 53 Plataformas



Ilustración 54 Brecha

La brecha está representada por un efecto de rayos; una vez el jugador cruza la brecha se acaba el juego y el jugador “gana” la partida. Este final del juego se ha diseñado teniendo en cuenta de que el objetivo principal del mismo es retener al

jugador para una futura versión. En la tercera isla el jugador encontrará edificaciones y animales, que no tienen nada que ver con lo que está acostumbrado, como arañas enormes o estructuras vikingas. Dando a entender que hay mucho más por explorar y descubrir en el mundo de BrokenLand. Una vez finaliza el juego, aparece una pantalla resumen donde se indica si el jugador ha acabado el juego (CONGRATULATIONS) o ha perdido (GAME OVER).



Ilustración 55 El jugador ha ganado



Ilustración 56 El jugador ha perdido

En esta pantalla, el jugador puede salir del juego, o volver al menú principal, de esta forma se da la posibilidad de realizar otra partida empezando de cero. El mundo de BrokenLand es peligroso y el jugador deberá enfrentarse a diferentes peligros, que pueden causar su muerte y perder la partida:

- **Pérdida de vida**: El jugador empieza con 5 puntos de vida que puede ir perdiendo a medida que juega, ya sea por falta de comida, explicado en el sistema de supervivencia, o porque ha sido atacado por animales agresivos y ha perdido toda la vida. Una vez el jugador pierde todos sus puntos de vida, se activa la animación de muerte, donde se representa al avatar cayendo al suelo desfallecido y se acaba la partida.



Ilustración 57 Muerte

- **Caída al vacío**: En un mundo fragmentado, uno de los mayores peligros es caer al vacío. Si el personaje del juego cae el vacío morirá

instantáneamente, generando tras de sí una explosión de partículas en forma de calavera, para dar a entender que se ha desintegrado.

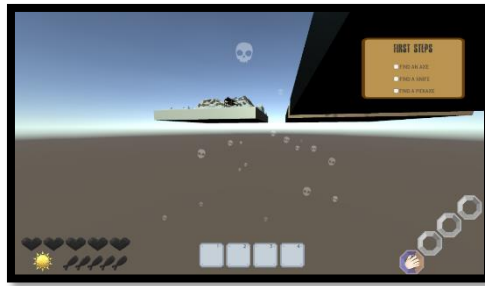


Ilustración 58 Caída al vacío

Menús:

Dentro del juego, el jugador tiene a su disposición tres menús que le permitirán realizar distintas acciones, ya sea para realizar alguna mecánica o para parar el juego.

- **Menú de pausa:** El jugador puede parar el juego en cualquier momento con la tecla P o ESC del teclado. Esto realiza las siguientes acciones a nivel del juego:
 - Parar el sistema de supervivencia.
 - Para el ciclo de día y noche.
 - Parar el funcionamiento de la regeneración de recursos.
 - Parar el movimiento de los animales.

Dentro del menú nos encontramos en que el jugador puede volver al menú principal y puede salir del juego desde este menú en cualquier momento.



Ilustración 59 Menú de pausa

- **Menú de inventario:** El jugador puede abrir el inventario del personaje con la tecla I, para saber que recursos ha recolectado. El inventario viene representado con un *grid* de 4x3 en los cuales en cada slot podemos encontrar la siguiente información:
 - SlotImage: Es una imagen del recurso que contiene ese slot del inventario, para que el jugador vea lo que es.
 - SlotValue: En la parte inferior del slot, podemos ver el número de unidades que tiene de ese recurso.

En caso de gastar todas las unidades de un recurso, se eliminará el recurso del inventario. Para conseguir que sea dinámico se ha creado un diccionario que permite buscar, añadir y quitar recursos del diccionario mediante una clave que es el nombre del recurso. Dicho diccionario se inicializa vacío al inicio del juego y va creciendo o disminuyendo a medida que el jugador obtiene o gasta los distintos recursos.



Ilustración 60 Menú de inventario

➤ **Menú de crafting:** Como en todo juego *survival* que se precie, en BrokenLands existe la posibilidad de realizar construcciones. Para ellos, se ha habilitado un menú con dos objetivos: revisar las necesidades de construcción de cada elemento y ofrecer la posibilidad de su construcción. Cada objeto a construir consta de un gran botón, que abarca tres indicadores:

- Slots de recursos: Indica, de igual forma que el inventario, los objetos que se necesitan y sus cantidades. Este slot compara las necesidades con el número de unidades que el jugador tiene de dicho objeto, para indicar de forma visual si el jugador posee las unidades necesarias (número verde) o no (número rojo).
- Slot de construcción: Muestra al jugador el objeto resultante de dicha construcción con una imagen del mismo.

Cuando el jugador hace click en el objeto, se inicia el sistema de construcción para poder construir el objeto deseado.



Ilustración 61 Menú de crafting

Inventory Info:

El jugador, debe saber en todo momento que objetos obtiene de los recursos. De esta manera se ha diseñado un sistema que detecta, cuando el jugador obtiene un recurso y lo muestra por pantalla, en la parte superior derecha. La información que se muestra es una imagen del objeto, el número de unidades que ha conseguido y el nombre del recurso.

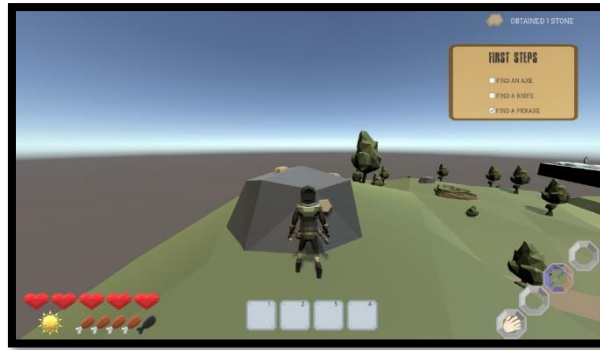


Ilustración 62 Inventory Info

4.8 Personaje

Como se ha descrito en el apartado [4.4](#), el modelo del personaje ha sido extraído de un paquete de Assets llamado Polygon Adventure Pack. El personaje controlado por el jugador, puede moverse por el escenario y realizar un numero extenso de acciones para interactuar con el mundo del juego, como recoger objetos o picar piedra.

Movimiento:

Para el movimiento básico del personaje y la cámara, se ha utilizado lo aprendido en la asignatura Unity 3D sobre los Standard Assets, pero debido a que ya no son compatibles, Unity recomienda usar el paquete Invector Free, que contiene un personaje con movimientos básicos como moverse y saltar. Se ha retocado para adecuarlo a las necesidades del juego, por ejemplo, para que se pare al realizar alguna acción o corregir comportamientos erráticos.

El jugador puede controlar al personaje mediante las teclas WASD del teclado y puede realizar saltos, para llegar a sitios elevados, pulsando la tecla SPACE.

Adicionalmente, el jugador puede mover la cámara (en tercera persona) para ver el entorno que le rodea.

Sistema de acciones:

Con tal de que el jugador pueda realizar distintas acciones, se ha desarrollado un sistema que, con la detección de dos elementos, determina que acción debe realizar.

El primer elemento que se detecta es la herramienta seleccionada, el jugador dispone de cuatro herramientas a su disposición y puede elegir que herramienta usar.

Con un sistema de colliders se detecta que recurso tiene delante el jugador y se guarda.

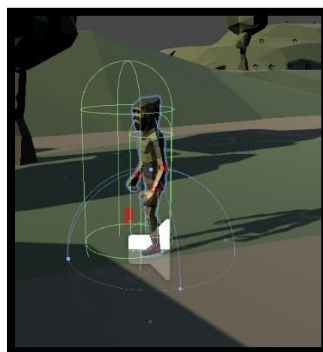


Ilustración 63 Colliders del avatar

Por lo tanto, en base a la herramienta elegida y al recurso que el avatar tiene delante, se determina la acción a realizar:

- **Coger**: Coge un objeto del suelo.
- **Cortar**: Corta con el hacha.
- **Picar**: Pica una piedra.
- **Atacar**: Ataca a un enemigo con el cuchillo.

Las animaciones se cambian en base a un número, que determina que animación realizar. Las animaciones han sido extraídas de Mixamo y combinadas en el Animator del jugador y su modelado (se han realizado algunos ajustes para que se vean correctamente las animaciones), para crear un sistema de acciones simple, que no tuviera mucha carga computacional.

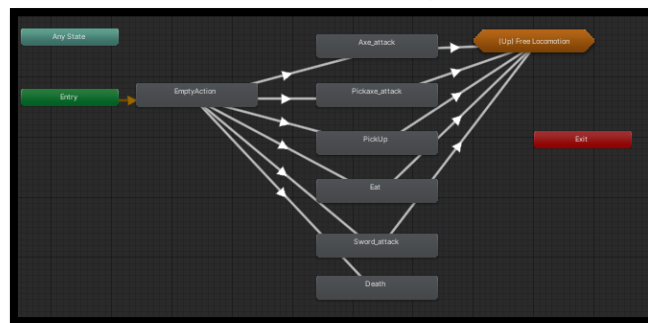


Ilustración 64 Diagrama de animaciones de las acciones

Efectos del personaje:

Uno de los objetivos del diseño del juego, es que sea auto explicativo, que el jugador obtenga mucho feedback para que él mismo determine qué está pasando. De esta forma, se ha evitado indicar al jugador, por ejemplo, que ha picado una piedra o ha comido una baya.

Con este objetivo en mente, se ha desarrollado una clase que se encarga de controlar los efectos del personaje, con dos enfoques:

- Audio: En base a la acción que realiza el jugador, se emite un sonido, limitado de forma corta para que el jugador sepa lo que ha hecho.
- Efecto visual: Del pack de Assets mencionado en el apartado [4.4](#) de efectos, se activa un efecto que se desactiva al poco tiempo, que lanza un sistema de partículas relativas a la acción que se ha hecho. Este efecto puede aparecer a dos alturas para representar dos acciones concretas:
 - Zona alta: Para picar piedra, talar árboles y atacar a enemigos altos.
 - Zona baja: Solo se activa al detectar enemigos bajos, como la serpiente.

Los diferentes efectos que se activan al ejecutar alguna acción son:

- Picar una piedra: Al picar una piedra, se emite un sonido de picar piedra y un efecto visual de partículas con piedra saltando, para simular el golpe del pico.

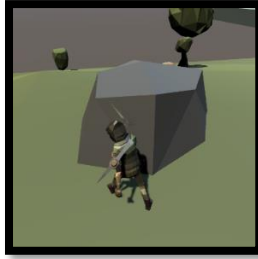


Ilustración 65 Efecto picar una piedra

- Talar un árbol: Al cortar un árbol con el hacha, aparece un efecto visual que simula el golpe y un efecto sonoro.

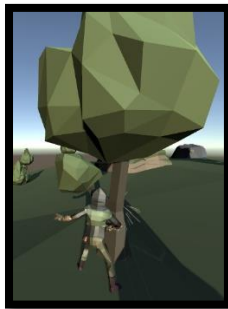


Ilustración 66 Efecto talar un árbol

- Atacar a un animal: Si el jugador ataca un animal, se escucha el sonido del cuchillo cortando y aparece un efecto de golpe en los animales.



Ilustración 67 Efecto de ataque

- Comer o coger un objeto: Si el personaje realiza la acción de comer o recoger algún objeto del suelo, se reproduce el sonido correspondiente.

4.9 Acciones

El jugador puede realizar diversas acciones en base a la herramienta que ha seleccionado en el cinturón de herramientas. Todas las acciones se pueden realizar pulsando la tecla de acción, la tecla E.

- **Coger:** No es necesaria ninguna herramienta y viene representada con una mano en el cinturón de herramientas. Permite coger cosas del suelo, como herramientas o las bayas de los arbustos.



Ilustración 68 Acción coger

- **Corte:** Únicamente se puede realizar si el jugador ha conseguido el hacha y selecciona el hacha en el cinturón de herramientas. Permite cortar los árboles, con el objetivo de derribarlos y obtener madera.

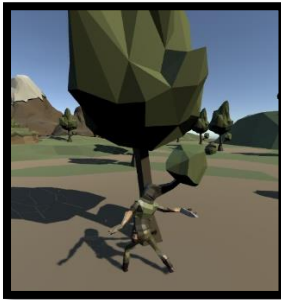


Ilustración 69 Acción cortar

- **Picar:** Únicamente se puede realizar si el jugador ha conseguido el pico y selecciona el pico en el cinturón de herramientas. Permite picar las piedras.

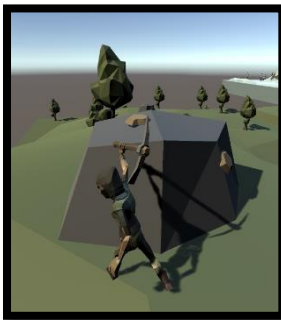


Ilustración 70 Acción picar

- **Ataque:** Únicamente se puede realizar si el jugador ha conseguido el cuchillo y selecciona el cuchillo en el cinturón de herramientas. Permite atacar a los enemigos. El ataque base del jugador es de 2.

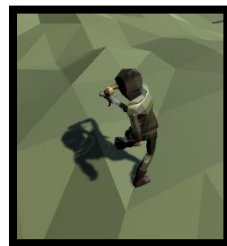


Ilustración 71 Acción ataque

- **Comer:** Existen dos formas de que el jugador pueda comer, a través del menú de inventario o a través del sistema de acceso rápido.

Para comer algún recurso desde el inventario, el jugador debe abrir el inventario, seleccionar la comida con el ratón y hacer click sobre ella. De esta forma, el personaje del juego se alimenta y recupera niveles de hambre.

En caso de querer comer de forma rápida, el jugador debe pulsar la tecla correspondiente al recurso a consumir del inventario de acceso rápido.



Ilustración 72 Paso 1: Coger el objeto



Ilustración 73 Paso 2-Inventario: Click en el objeto a comer



Ilustración 74 Paso 2-Acceso Rápido: Pulsar tecla asociada



Ilustración 75 Paso 3: Comer

4.10 Construcción

En la mayoría de juegos del género *survival* el jugador puede unificar recursos para crear nuevos. En BrokenLand, el jugador puede realizar construcciones para poder realizar algunas acciones específicas. Estas construcciones no son necesarias para avanzar en el juego, sino que se han planteado como objetivos opcionales para el jugador.

Sistema de construcción:

Para el diseño del sistema de construcción se han tomado referencias de diferentes juegos con entorno 3D para realizar las construcciones.

El sistema de construcción se inicia abriendo el menú de crafting para que el jugador pueda ver que recursos necesita recolectar para cada construcción.



Ilustración 76 Menú de crafting con indicadores

En la imagen superior se pueden ver varios indicadores visuales que informan al jugador si puede o no, realizar la construcción:

- Cada slot de recurso indica el número total de unidades que son necesarias de dicho recurso. También indica en la parte izquierda, cuantas unidades posee el jugador actualmente.
- En caso de tener las unidades requeridas, el número de las unidades que el jugador tiene en el inventario, aparece en color verde. En caso contrario, aparece en rojo.
- Si el jugador tiene todos los elementos necesarios para construir el objeto, el botón de la construcción aparece en color. Si no dispone de los recursos necesarios, aparece en gris.

Una vez que el jugador hace click en el objeto a construir (en color), aparece en frente del personaje la sombra del objeto. Esta sombra sirve para elegir dónde colocar el objeto dentro del mapeado de las islas.



Ilustración 77 Sombra al construir

Esta sombra, actúa también para verificar si el objeto puede ser construido o no. Los objetos no pueden construirse atravesando otros objetos o atravesando alguna montaña. La sombra aparece en verde si el objeto puede construirse y en rojo si el objeto no puede construirse.



Ilustración 78 Construcción imposible

Este tiempo de gracia, entre que el jugador manda la orden de construcción y se realiza la construcción sirve para que el propio jugador decida donde colocar la construcción. Haciendo click con el ratón, se realiza la construcción.



Ilustración 79 Hoguera construida

Hoguera:

El primer objeto que puede construir el jugador es una hoguera. Esta construcción se ha pensado para que actúe como punto de luz en la noche y así indicar zonas por donde el jugador ya ha pasado. Está pensado también para que, en una versión futura, exista un indicador de temperatura y el jugador pueda calentarse en la hoguera.

La hoguera detecta el momento del día y en caso de ser de noche, se activa, ofreciendo un efecto visual de fuego y un punto de luz.



Ilustración 80 Hoguera

Cama:

Una de las construcciones más importantes del género *survival* es la cama. Es un objeto que normalmente permite al jugador pasar toda la noche, debido a que se ve poco y en algunos juegos aumenta la dificultad. En BrokenLand una vez la

cama ha sido construida, el jugador puede dormir en ella pulsando la tecla de acción cerca de ella.

Se iniciará la animación de dormir y el tiempo se verá acelerado. Una vez se haga de día de nuevo, el personaje se levantará de la cama y podrá seguir su aventura. Hay que tener cuidado, porque, aunque pase el tiempo, la pérdida de niveles de vida y hambre, también se ve incrementada.



Ilustración 81 Cama construida



Ilustración 82 Jugador durmiendo

4.11 HUD

En el HUD aparece toda la información relevante que el jugador necesita para poder entender el juego y saber que vitalidad tiene con tal de sobrevivir.

En el HUD, vienen representados los siguientes elementos:

- **Vida:** Mediante unos iconos de corazones se representa la vida del jugador. Si la pierde toda, muere y se acaba el juego. El máximo de la vida es de 5 corazones.



Ilustración 83 Corazones indicando la vida

- **Hambre:** Representada por unos muslos de pollo, una pequeña referencia a Minecraft, está mostrada el hambre del jugador. Si el jugador pierde toda su hambre, pasa a estado hambriento y empieza a perder vida progresivamente.



Ilustración 84 Muslos de pollo representando el hambre

- **Tiempo:** Actualmente no es un elemento del HUD que aporte mucha información, dado que solo indica si es de día o de noche mediante unos iconos de un sol y una luna; pero en un futuro, si se quieren dar indicaciones al jugador sobre el tiempo, por ejemplo, que va a llover; se indicará en este elemento.



Ilustración 85 Indicador visual del sol



Ilustración 86 Indicador visual de la luna

- **Cinturón de herramientas:** El juego está basado en las acciones que te permiten realizar distintas herramientas. Estas herramientas están en el cinturón que se puede ver en el HUD, representados por pequeños iconos que contienen (si el jugador las ha conseguido) las distintas herramientas. Es posible cambiar de herramienta con la rueda del ratón. La localización de las herramientas es: Hacha en la colina de las setas, Cuchillo cerca de los montículos en la esquina opuesta al final del juego y el pico está en una piedra gigante al lado de la gran montaña.



Ilustración 87 Cinturón de herramientas

- **Acceso rápido:** Con el objetivo de que el jugador, pueda comer de forma rápida, se ha añadido un sistema de acceso rápido en la parte inferior central del HUD. Actúa de igual forma que el inventario, por lo tanto, es dinámico, a medida que el jugador obtenga comida, aparecerá en las teclas de acceso rápido. Si el jugador pulsa la tecla, se consumirá la comida correspondiente a dicha tecla y se restará el recurso utilizado. Filtra los materiales, para solo mostrar aquello que pueden ser usados de forma rápida, por lo tanto, la comida.



Ilustración 88 Acceso rápido

4.12 Entorno

En BrokenLand el mundo ha cambiado y ya no existe la tierra como tal. El mundo ahora está formado por islas flotantes que albergan misterios y peligros.

Existen tres islas, totalmente diferentes entre sí y con biomas diferentes. Los biomas se han diseñado para que el jugador encuentre entornos diferentes y no se haga monótona la exploración. Desde islas con mucha vegetación a páramos nevados. Con los biomas diseñados, en una futura versión se crearán más islas de cada bioma, de esta forma se pueden encontrar formaciones diferentes de las islas de un mismo bioma.

Isla inicial:

En esta isla nos encontramos con multitud de árboles, dando un aspecto de bosque denso. La tonalidad de la isla es verde para dar sensación de calidez y de no peligro.

Relacionado con el peligro, en esta isla no se ha añadido ningún enemigo para que así el jugador se familiarice con las mecánicas y una vez ya ha aprendido todo, pueda salir al exterior y enfrentarse a los peligros que le esperan.

Al ser la isla inicial, se ha diseñado con la intención de que sea un punto de referencia para el jugador, un lugar seguro. De esta forma es la isla con más comida, debido a que los climas de las otras islas no permiten que florezcan arbustos o árboles frutales. De esta forma se crea un flujo de backtracking, donde incitamos al jugador a volver al lugar seguro que es la primera isla para obtener más comida.

En esta isla es donde se pueden encontrar las tres herramientas del juego, escondidas, que el jugador deberá buscar nada más empezar la partida.



Ilustración 89 Isla inicial 1

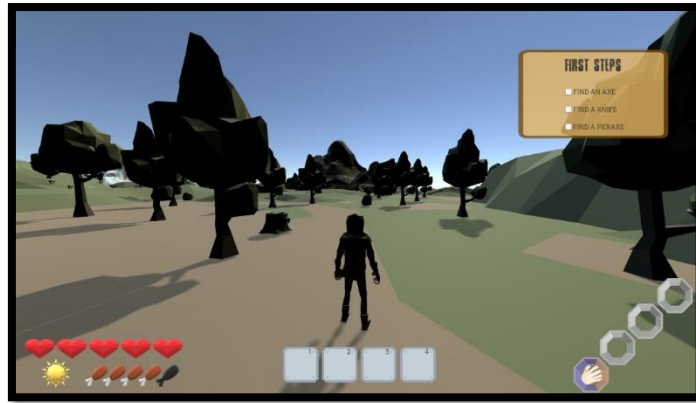


Ilustración 90 Isla Inicial 2

Segunda isla:

El diseño de la segunda isla ha partido de tres conceptos:

- Nuevos recursos: En esta isla el jugador puede encontrarse con llamas, que al derrotarlas puede conseguir piel. Esta piel sirve para la construcción de la cama.
- Primeros enemigos: En esta isla el jugador se encontrará con sus primeros enemigos que deberá derrotar si quiere los recursos o en el caso de encontrarse con un enemigo agresivo, deberá derrotarlo para sobrevivir.
- Pocos recursos: La idea de esta isla es que sea una isla de paso, que el jugador no pueda quedarse mucho, debido a que no hay comida.

El entorno se ha diseñado simulando una zona desértica con árboles sin hojas, árboles muertos. La isla presenta formaciones rocosas y el punto más alto del juego, una montaña enorme que se puede ver desde cualquier punto del mundo. La decisión de crear una formación montañosa tan grande es para atraer al jugador, al poderse ver desde cualquier punto del mapa, el jugador ve algo tan enorme que le atrae a la exploración.

Los primeros enemigos que se encuentra el jugador en esta isla son las llamas (pasivas) y las serpientes (agresivas).



Ilustración 91 Segunda Isla 1



Ilustración 92 Segunda Isla 2

Isla destino:

El diseño original de BrokenLand no comprendía un final del juego como tal, pero debido a las limitaciones, se ha tenido que añadir una isla en donde se puede terminar el juego.

La última isla se ha creado con el objetivo principal de dejar al jugador con ganas de más, creando una sensación de misterio en esta última isla. Debido a que en ella hay construcciones que no parecen de una época actual, estatuas e incluso criaturas que no se han visto antes, como arañas enormes.

Como objetivo principal del juego, el jugador puede encontrar en la isla destino la brecha para finalizar el juego, después de realizar una serie de saltos entre plataformas hasta llegar al antiguo campamento vikingo. Además, la isla destino esconde algún secreto, como, por ejemplo, estructuras escondidas por las montañas o lo que parece ser una especie de estructura semejante a una casa. En esta isla, nos encontramos con el primer efecto del clima, que da un toque visual mucho más atractivo al entorno, debido a que no para de nevar. La nieve, para no estar constantemente consumiendo recursos, solo funciona cuando el jugador está en la isla destino.

El jugador puede encontrar todos los recursos principales del juego en esta isla, pero de forma limitada. Como puede ser la madera, la piedra y algo de comida en las setas.

Los enemigos que se encuentra el jugador en esta isla son los caballos salvajes (pasivos) y las enormes arañas (agresivas).

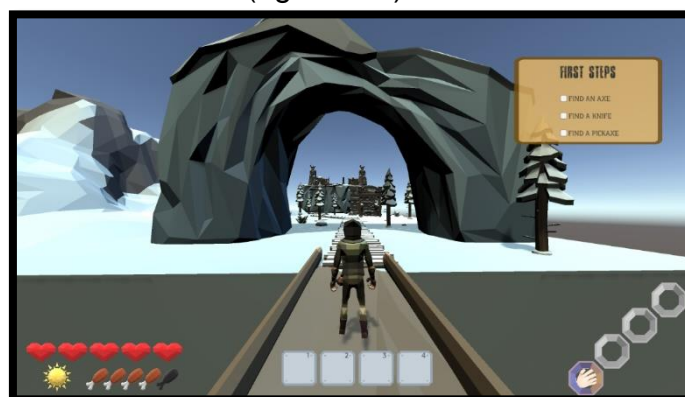


Ilustración 93 Isla Destino 1

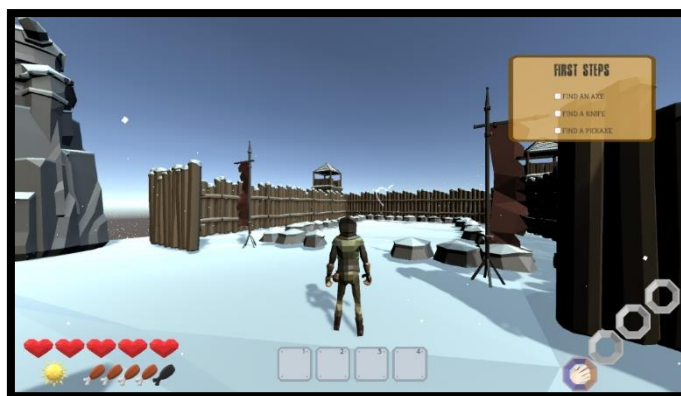


Ilustración 94 Isla Destino 2

Sistema de conexión de islas:

En BrokenLand el jugador se encuentra con un mundo fragmentado que debe unir para poder sobrevivir. Para la conexión de las diferentes islas se han diseñado unas construcciones en espacios fijos dentro del mundo del juego.

Inspirado en la saga del Archivo de las Tormentas de Brandon Sanderson, la conexión de las islas flotantes se realiza mediante puentes.

Para identificar los puentes que el jugador puede construir, lo primero que verá el jugador son unas pequeñas estructuras de madera que delimitan el puente e indican hacia donde se puede construir.



Ilustración 95 Indicador de puente

Cuando el jugador se acerca, aparece un indicador que informa de que recursos necesita y cuantas unidades son necesarias de cada uno. Siguiendo el mismo patrón que el menú de crafting, el cartel aparecerá de color gris si el jugador no posee los recursos necesarios y de color naranja si el jugador puede construir el puente.



Ilustración 96 Indicador flotante de puente

En el cartel se indica al jugador que debe pulsar la tecla C para empezar la construcción del puente. Lo primero que verá el jugador es un efecto de humo

que indica que el puente se está construyendo. Se evita que, durante su construcción, el jugador no pueda iniciar otra construcción.

El jugador ve como las pequeñas estructuras que delimitan el puente dejan paso a un puente de madera y piedra construido completamente. El jugador puede pasar en cualquier momento durante su construcción, pero se arriesga a caer al vacío.

Una vez ha finalizado la construcción del puente, el jugador ya puede viajar libremente entre islas.

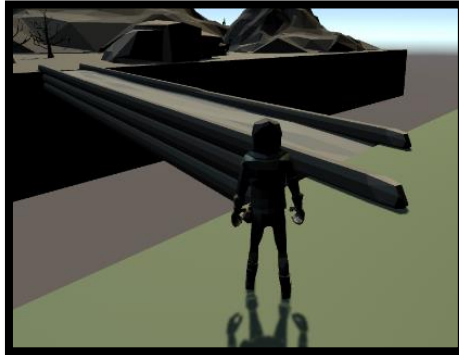


Ilustración 97 Puente construido 1

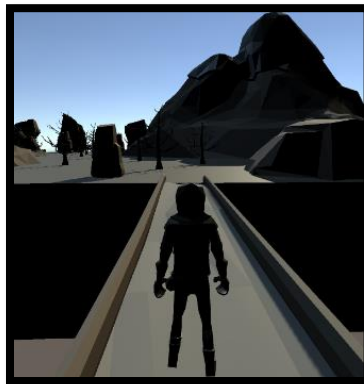


Ilustración 98 Puente construido 2

En total existen dos puentes en el mundo de BrokenLand, que se consideran como objetivos principales para poder finalizar el juego, debido a que, si el jugador no los construye, no podrá viajar al resto de islas.

4.13 Recursos

A lo largo de su aventura, el jugador deberá recolectar recursos para poder sobrevivir y poder construir los diferentes puentes que le ayudarán a llegar al final del juego.

Existen de diferentes tipos, algunos que alimentan al jugador, por lo tanto, contribuyen al objetivo principal del juego que es sobrevivir; y otros que son necesarios para realizar la construcción de los puentes y poder llegar al final.

Los diferentes recursos presentes en el mundo de BrokenLand son:

- **Madera:** Se obtiene cortando los árboles, pero antes el jugador deberá conseguir el hacha. Los árboles tienen una resistencia de 10, una vez agotamos esta resistencia, se activa una animación (creada por mí a través del Animator de Unity) que hace que el árbol caiga. Una vez en el suelo, desaparece, el jugador obtiene 1 unidad de madera y la puede visualizar en el inventario.

Existen diferentes tipos de árbol, uno diferente para cada isla: árbol con hojas, árbol muerto y árbol nevado. Cada uno tiene una animación ligeramente diferente para que se vea correctamente el árbol en el suelo, debido a que no tiene los mismos tamaños ni estructuras.



Ilustración 99 Árbol Isla Inicial



Ilustración 100 Árbol Segunda Isla



Ilustración 101 Árbol Isla Destino

A parte de los árboles específicos de cada isla, el jugador puede obtener madera cortando algún árbol frutal, pero claro, perdería una fuente de comida que se regenera de forma constante.

- **Piedra:** Se obtiene picando las menas, pero antes debemos conseguir el pico. Las menas tienen una resistencia de 10 y se pueden obtener hasta 4 piedras. Cada piedra se obtiene al superar un 25% de la resistencia y para indicar al jugador de que se ha conseguido, un elemento de la piedra desaparece (un trozo de piedra).



Ilustración 102 Mena de piedra

Una vez se han picado todas las piedras de la mena, la piedra desaparece.

- **Bayas:** Se obtiene de los arbustos de bayas y es el único recurso para el que no necesitamos herramientas, se cogen con la mano. Los arbustos tienen una resistencia de 6 y se pueden obtener hasta 6 bayas. Cada baya se obtiene al superar un 20% de la resistencia y para indicar al jugador de que se ha conseguido, una baya del arbusto desaparece. El arbusto no desaparece, debido a que las bayas se regeneran. Las bayas alimentan al jugador en una unidad de hambre.



Ilustración 103 Arbusto de bayas

- **Setas:** Para que existiera alguna fuente de comida que encajara con el bioma de la isla de destino (nevado) se ha creado una fuente de alimento que sobreviva a ese clima como son las setas. Las setas se pueden recoger del suelo con la mano y producen una unidad de hambre al jugador. Las setas no solo están presentes en la última isla, también pueden encontrarse en una colina en la primera isla, debido a que la primera isla es introductoria al juego y de esta forma, se le muestra al jugador otra fuente de comida.

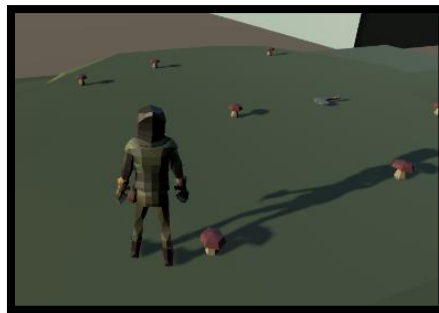


Ilustración 104 Setas

- **Manzanas:** En la primera isla, existe multitud de formas de obtener comida, para que el camino no se le haga demasiado duro al jugador, debido a que el objetivo de BrokenLand es ser un juego pacífico y nada estresante. Con este objetivo en mente se ha diseñado un tipo de árbol que produce fruta de forma constante, a menos que sea talado. Se ha creado un manzanero que produce manzanas.



Ilustración 105 Manzanero

El modelo del árbol es ligeramente diferente a los que el jugador puede encontrar en la primera isla, de esta forma se consigue que el jugador deba mirar a los árboles para saber si son frutales o no.

Los manzaneros tienen cuatro puntos donde crecen las manzanas, en sus ramas. Las manzanas inicialmente no están maduras y por eso están sujetas al árbol y el jugador no puede recogerlas.

Después de un tiempo, la manzana madura y cae al suelo, así el jugador puede recogerla. Las manzanas son la mejor comida que se puede encontrar en BrokenLand y proporcionan dos puntos de hambre. Se ha diseñado de esta forma debido a que las manzanas tienen dos hándicaps y no son tan sencillas de conseguir:

- Tardan un tiempo en madurar y caer al suelo.
- Tardan bastante tiempo en regenerarse.

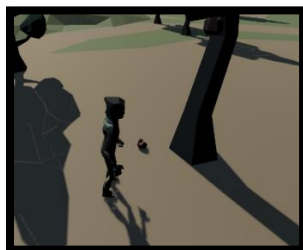


Ilustración 106 Caída de fruta

El efecto de caída del árbol se consigue con dos elementos. Por un lado, las manzanas se crean de forma que son kinemáticas por lo tanto no les afecta la gravedad. Una vez pasa el tiempo de maduración, pasan a ser no kinemáticas, la gravedad les afecta y caen. Además, el script encargado de la caída de las manzanas, mediante una función aleatoria, selecciona que manzana va a caer, por lo tanto, no siempre caerán en el mismo orden.

Sistema regeneración de recursos:

En la todos los referentes del género *survival* existe una norma no escrita, de que los recursos deben regenerarse. De esta forma se evita que el jugador se quede sin los recursos necesarios para conseguir los objetivos, ya sea porque los ha gastado o porque le faltan recursos adicionales. Con este objetivo en mente se ha diseñado un sistema de regeneración con tres características fundamentales:

- Las setas no se regeneran: Las setas sirven específicamente para que en la última isla el jugador encuentre comida. Como en dicha isla, nieva sin

parar y en la isla inicial no llueva (que podría hacer que crecieran las setas); se ha decidido que no se regeneren.

- La regeneración no se produce al estar el jugador cerca: Para evitar situaciones sin control, se ha añadido un collider a cada objeto que se debe regenerar. De esta forma se verifica que el jugador no está cerca, para poder regenerar el objeto.
- Diferentes tiempos de regeneración dependiendo de la dificultad: Después de varias pruebas, se ha determinado que, para balancear el juego, cada nivel de dificultad tenga unos tiempos de regeneración diferentes.

Dificultad	Elemento	Tiempo
Fácil	Piedra	1 min.
Fácil	Árbol	1 min.
Fácil	Baya	1 min.
Fácil	Manzana	2 min.
Normal	Piedra	5 min.
Normal	Árbol	5 min.
Normal	Baya	5 min.
Normal	Manzana	10 min.
Difícil	Piedra	10 min.
Difícil	Árbol	10 min.
Difícil	Baya	10 min.
Difícil	Manzana	20 min.

Para conseguir el efecto de destrucción y regeneración, lo que se ha realizado es que todos los objetos se desactiven al ser destruidos; y se reseteen y se activen al regenerarse.

El elemento que tiene una regeneración adicional es el manzanero. El árbol frutal se regenera a sí mismo, si el jugador tala el árbol. Pero también tiene una regeneración en las manzanas que produce.

Una vez una manzana cae, manda una orden al nodo del árbol de donde ha caído para indicar que debe regenerarse. El nodo inicia una cuenta atrás para la regeneración de la manzana, una vez pasa ese tiempo, la manzana vuelve a aparecer en el nodo.



Ilustración 107 Múltiples caídas de fruta

Como las manzanas caídas, no se pretende que desaparezcan, constituyen un objeto por si sola, como manzanas maduras. De esta forma se evita que el conjunto de manzanas desaparezca en caso de que el jugador decida talar el árbol. Si el árbol es talado, las manzanas aún en el árbol desaparecen, debido a que no pueden ser comidas por el jugador por no estar maduras.



Ilustración 108 Manzanero talado

Las bayas de los arbustos se regeneran por si solas, por lo tanto, cuando el jugador coge una de las bayas, se iniciará el sistema de regeneración para volver a florecer.

Las menas de piedra no es un elemento que en la naturaleza vuelva a crecer, pero como no existe otro método de obtención de piedra y tomando otros juegos del género como referencia (Conan Exiles) se ha decidido que se regenere después de un tiempo. El jugador debe destruir la mena por completo para que se inicie el proceso de regeneración de la mena entera.

4.14 NPC e IA

En el diseño inicial de BrokenLand, no tenían cabida los NPCs, debido a que el objetivo principal del juego no tiene por qué ser enfrentarse a los enemigos. Se han añadido al juego para dar más vida al mundo.

Con este objetivo en mente, no se ha añadido ninguna recompensa por derrotarlos (más allá de la piel de la llama), sino que si el jugador decide enfrentarse a ellos y atacarlos que sea para defenderse. Esta decisión de diseño se ha tomado en concordancia con la idea inicial de hacer un juego *survival* pacífico, sin muchos peligros.

Es más, el jugador puede terminar el juego sin tener que derrotar a ningún enemigo, debido a que el único enemigo que ofrece alguna recompensa es la llama y sirve para construir la cama, que no aporta gran valor en el objetivo de llegar a la isla de destino.

Tipos de enemigos:

Como el objetivo principal de la inclusión de enemigos en BrokenLand es para hacer el mundo más vivo, se han separado los enemigos para que cada uno tenga un comportamiento diferente. Los comportamientos de los enemigos son diferentes entre sí, pero se pueden agrupar en dos grandes grupos:

- **Agresivo:** Estos enemigos deambulan por la isla donde viven, sin atacar a nadie, pero cuando detectan al jugador, se sienten amenazados y atacan.
- **Pasivo:** Son animales que no tienen ninguna intención de pelear, sino que pasean por la isla donde viven, con ningún objetivo más que vivir. Estos animales son los más fáciles de derrotar, debido a que no ofrecen resistencia.

Todos los animales en BrokenLand tienen alguna peculiaridad, para que el jugador se pare a estudiarlos antes de atacar, en caso de querer enfrentarse a ellos. Las diferencias entre ellos, van más allá de un cambio de características, sino que tienen comportamientos y reacciones diferentes entre sí.

Enemigo pasivo 1, la llama:

El primer animal que se encuentra el jugador es la llama. Un animal acostumbrado a vivir en montañas y por ese motivo se decidió añadirlo como animal pasivo en la segunda isla, que presenta grandes montañas.

La llama es un animal fácil de derrotar, dado que no ataca, y de un solo ataque el jugador puede derrotarla. Como recompensa ofrece una unidad de piel que puede utilizarse para la creación de la cama.



Ilustración 109 Llama

La llama tiene un sistema de navegación que le permite deambular por la isla. El sistema de navegación de la llama es cíclico con diferentes fases:

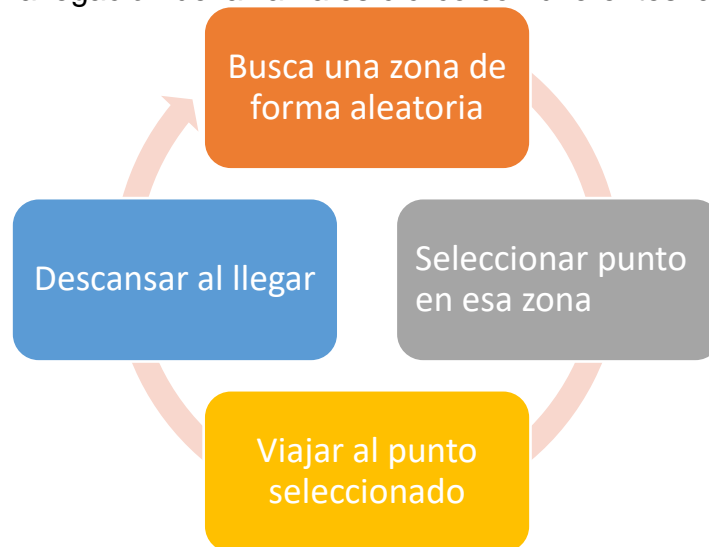


Ilustración 110 Diagrama IA Llama

- Paso 1: Hay un conjunto de zonas específicamente para las llamas, dentro de ese conjunto, se selecciona una a la que viajará la llama.



Ilustración 111 Zonas de patrulla de Llama

- Paso 2: Cada zona está diseñada como un rectángulo 3D y la llama selecciona un punto aleatorio dentro de ese rectángulo, como punto de destino.
- Paso 3: La llama se desplaza al punto elegido como punto de descanso.

- Paso 4: Descansa en el punto elegido durante un tiempo antes de emprender su camino.
- Paso 5: Se repite el ciclo y vuelve a buscar una zona.

La búsqueda de zonas se ha diseñado de forma aleatoria porque así no todas las llamas van al mismo sitio, sino que van a puntos diferentes y separadas entre sí.

Las animaciones de la llama se han diseñado partiendo de animaciones de otros animales, si bien es cierto que se podría elegir otro animal diferente que ya tuviera las animaciones, se ha decidido incorporar la llama por ser un homenaje a una persona de mi entorno.

Las estadísticas de la llama son:

Estadística	Valor
Vida	1
Ataque	0

El jugador se encontrará con animaciones diferentes en base a los estados de la llama, así como cuando la llama muere:



Ilustración 112 Llama descansando



Ilustración 113 Llama caminando

Enemigo agresivo 1, la serpiente:

Para la zona del desierto, la zona árida de la segunda isla, se ha incorporado un enemigo peligroso. No es un peligro real para el jugador, dado que no es necesario eliminarlo para avanzar, pero en caso de encontrarse con varias serpientes, el jugador puede llegar a perder vida y tener que huir a un lugar seguro para recuperar vida.

La serpiente es un enemigo pequeño, quizás difícil de ver desde lejos, eso hace que pueda llegar a sorprender al jugador si no es cauteloso.

La inteligencia artificial de la serpiente se ha basado en el patrón de máquina de estados. Los estados definidos para la serpiente son:

- Patrulla: La serpiente viaja de una zona a otra, siempre pendiente de si detecta una amenaza como es el jugador.
- Alerta: Una vez detecta al jugador porque está cerca, rota sobre si misma para encararse al enemigo, en este caso, al jugador.
- Ataque: Persigue al jugador atacándolo hasta que el jugador sale de su alcance.

El estado de la serpiente puede cambiar en cualquier momento, siempre en base a si ha detectado al jugador en los alrededores.

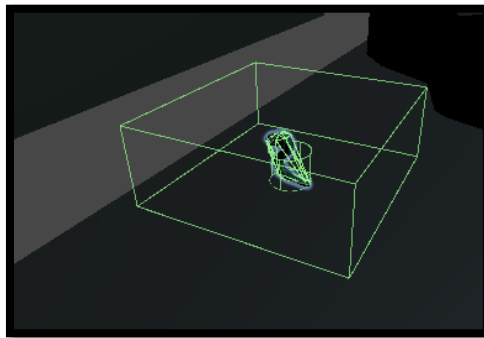


Ilustración 114 Área de detección de serpiente

El objeto de la serpiente tiene un collider que representa el rango de detección de la serpiente. En este caso se ha diseñado un rango pequeño, para que no sea una amenaza para el jugador en la segunda isla.

El diagrama que define la máquina de estado de la IA de la serpiente es el siguiente:

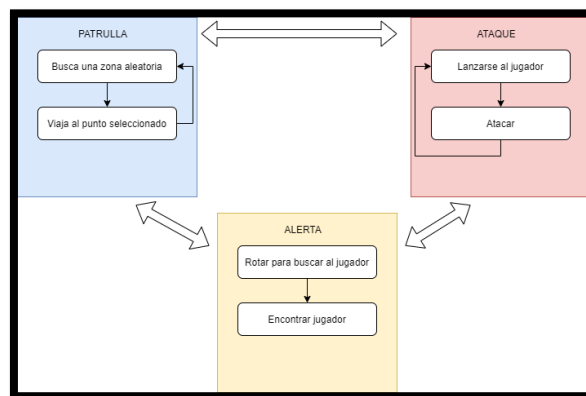


Ilustración 115 Diagrama IA Serpiente

El flujo de actuación de la serpiente en cada estado es el descrito a continuación:

- Patrulla:
 - Busca una zona aleatoria dentro de las designadas para las serpientes.
 - Viaja hasta el punto seleccionado.
 - Vuelve a buscar una zona aleatoria, para repetir el ciclo de patrulla.

- Alerta:
 - La serpiente rota sobre si misma para detectar donde está el jugador. Esto se consigue con un raycast que se lanza desde la boca de la serpiente y rota junto a la serpiente.
 - Si el raycast lanzado por la serpiente encuentra al jugador pasará al estado de ataque. En caso contrario continuará con el estado de patrulla.

- Ataque:
 - El ataque se basa en dos aspectos: que el jugador está dentro del rango y que la serpiente puede llegar a tocar al jugador.
 - Una vez se ha detectado al jugador, la serpiente se mueve hacia el jugador para proceder al ataque.

- Una vez la serpiente toca al jugador, le hace daño.

La serpiente tiene una resistencia de 1 corazón, por lo tanto, con un ataque la derrotamos. Si que es cierto, que, al tener poco rango de detección, el jugador podrá huir fácilmente en caso de encontrarse con una serpiente que le haya detectado.

Por lo tanto, si la serpiente deja de detectar al jugador en el rango de detección, pasará al estado de alerta para buscarlo y posteriormente al estado de patrulla si no lo encuentra.

Las estadísticas de la serpiente son:

Estadística	Valor
Vida	1
Ataque	1

El jugador se encontrará con animaciones diferentes en base a los estados de la serpiente, así como cuando la serpiente muere:

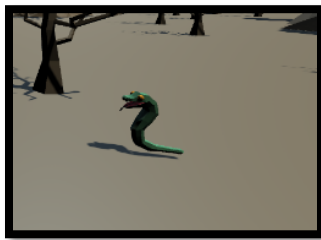


Ilustración 116 Serpiente patrullando



Ilustración 117 Serpiente en alerta



Ilustración 118 Serpiente atacando

Enemigo pasivo 2, el caballo:

Para el diseño del segundo animal se ha pensado en algo visualmente bonito y que aporte un punto de elegancia a la tercera isla con el bioma nevado. De esta forma se ha decidido incorporar caballos, debido a que tendrán un comportamiento diferente a la llama y puede ser interesante para el jugador verlos correr por las montañas nevadas.

Los caballos no son fáciles de derrotar debido a que tienen una mecánica de huida al ser golpeados. Cuando un caballo es golpeado aumenta su velocidad para huir del jugador. Como recompensa ofrece tres pieles debido a que es un animal grande.

El caballo tiene un sistema de navegación que le permite deambular por la isla.

El sistema de navegación del caballo es cíclico con diferentes fases:

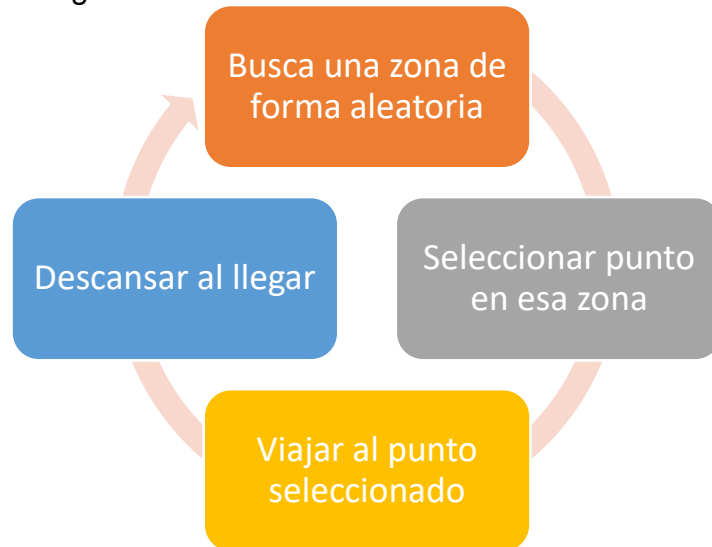


Ilustración 119 Diagrama IA Caballo

- Paso 1: Hay un conjunto de zonas específicamente para los caballos en la tercera isla, dentro de ese conjunto, se selecciona una a la que viajará el caballo.

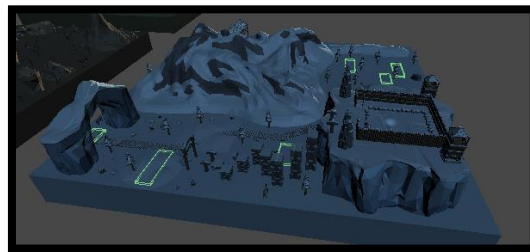


Ilustración 120 Áreas de patrulla de caballo

- Paso 2: Cada zona está diseñada como un rectángulo 3D y el caballo selecciona un punto aleatorio dentro de ese rectángulo, como punto de destino.
- Paso 3: El caballo se desplaza al punto elegido como punto de descanso.
- Paso 4: Descansa en el punto elegido durante un tiempo antes de emprender su camino.
- Paso 5: Se repite el ciclo y vuelve a buscar una zona.

La búsqueda de zonas se ha diseñado de forma aleatoria porque así no todos los caballos vayan al mismo sitio, sino que van a puntos diferentes y separados entre sí.

La peculiaridad del caballo respecto a la llama, es que el caballo se asusta al ser golpeado. El caballo resiste hasta dos ataques del jugador y por ese motivo se ha diseñado de tal forma que huya del jugador, haciendo más divertido el hecho de perseguirlo para cazarlo.

Una vez el caballo es golpeado, el caballo pasa a un estado de miedo y aumenta su velocidad. Con este cambio de velocidad, también cambia su animación para simular que corre; de esta manera el jugador lo verá alejarse mucho al atacarlo por primera vez y tendrá que perseguirlo si quiere derrotarlo.

Las estadísticas del caballo son:

Estadística	Valor
Vida	6
Ataque	0

Las animaciones del caballo han sido una mezcla entre usadas las del pack de assets, recreadas a partir de otros animales o creadas desde cero.

El jugador se encontrará con animaciones diferentes en base a los estados del caballo, así como cuando el caballo muere:

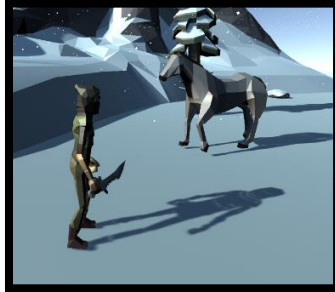


Ilustración 121 Caballo caminando



Ilustración 122 Caballo descansando

Enemigo agresivo 2, la araña gigante:

Con el objetivo en mente de introducir un enemigo que no se espere el jugador, se ha introducido como segundo enemigo agresivo, una araña gigante. De esta forma, el jugador se preguntará por qué existen arañas tan grandes que no existen en el mundo real y que hacen ahí. La idea de la última isla de BrokenLand es que el jugador se quede con ganas de más, de saber que ha pasado con el mundo y realmente saber por qué en la última isla hay criaturas extrañas como las arañas gigantes o la ambientación vikinga.

La inteligencia artificial de la araña gigante se ha basado en el patrón de máquina de estados. Los estados definidos para la araña son:

- Patrulla: La araña viaja de una zona a otra, siempre pendiente de si detecta una amenaza como es el jugador.
- Alerta: Una vez detecta al jugador porque está cerca, rota sobre si misma para encararse al enemigo, en este caso, al jugador.

- Ataque: Persigue al jugador atacándolo hasta que el jugador sale de su alcance.

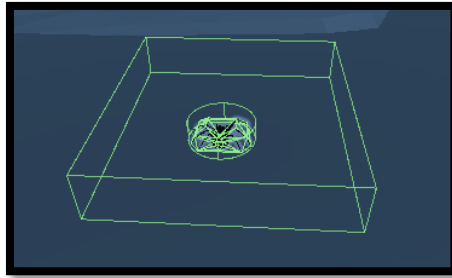


Ilustración 123 Área de detección de araña

El rango de detección es mucho más amplio que el de la serpiente y en muchas ocasiones no te esperas un ataque, por lo tanto, hay que estar muy pendiente de estos enemigos y tener cuidado.

El diagrama que define la máquina de estado de la IA de la araña es el siguiente, similar al de la serpiente:

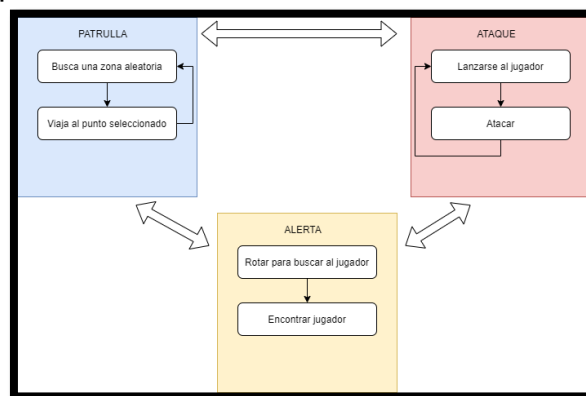


Ilustración 124 Diagrama IA Araña

El flujo de actuación de la serpiente en cada estado es el descrito a continuación:

- Patrulla:
 - Busca una zona aleatoria dentro de las designadas para las arañas.
 - Viaja hasta el punto seleccionado.
 - Vuelve a buscar una zona aleatoria, para repetir el ciclo de patrulla.
- Alerta:
 - La araña rota sobre si misma para detectar donde está el jugador. Esto se consigue con un raycast que se lanza desde la boca de la araña y rota junto a la araña.
 - Si el raycast lanzado por la araña encuentra al jugador pasará al estado de ataque. En caso contrario continuará con el estado de patrulla.
- Ataque:
 - El ataque se basa en dos aspectos: que el jugador está dentro del rango y que la araña puede llegar a tocar al jugador.
 - Una vez se ha detectado al jugador, la araña se mueve hacia el jugador para proceder al ataque.
 - Una vez la araña toca al jugador, le hace daño.

La araña tiene una resistencia de 4 corazones, por lo tanto, es un enemigo que mínimo tendremos que atacar dos veces para derrotarlo. Al tener tanto rango de detección, es el enemigo más peligroso del juego.

Adicionalmente, la araña hace dos corazones de daño al atacar, por lo tanto, el jugador si es atacado por una araña, tendrá que reponerse antes de poder seguir, por si acaso vuelve a ser atacado.

Por lo tanto, si la araña deja de detectar al jugador en el rango de detección, pasará al estado de alerta para buscarlo y posteriormente al estado de patrulla si no lo encuentra.

Las estadísticas de la araña son:

Estadística	Valor
Vida	4
Ataque	2

El jugador se encontrará con animaciones diferentes en base a los estados de la araña, así como cuando la araña muere:



Ilustración 125 Araña patrullando



Ilustración 126 Araña atacando

4.15 Descripción de clases

Cada clase utilizada en el desarrollo de BrokenLand está comentada en el mismo script. Están divididas en las carpetas en las que están organizadas dentro del repositorio. Las clases desarrolladas han sido:

Global:

- DayNightManager: Clase encargada de gestionar la rotación del sol para simular el ciclo día y noche.
- EndGameZone: Gestiona la finalización del juego, al llegar el jugador a la brecha temporal.
- EndMenuManager: El mensaje y menú final del juego, se gestionan desde esta clase.
- FadeManager: Administra la transición de escenas, a modo de fade-in / fade-out.

- GameOverWorldManager: Clase transversal que se encarga de la mecánica de supervivencia y de la gestión de todo el juego, como puede ser la apertura de menús.
- GameStats: Todas las variables globales se guardan en esta clase, para ser consultadas desde todas las demás clases.
- Menu: Contiene y gestiona las diferentes opciones de los menús principales que aparecen en BrokenLand.
- SnowManager: En la tercera isla, esta clase es requerida para la activación y desactivación de la nieve.
- Utils: Las funciones que son utilizadas por diversas clases, están contenidas en la clase Utils.

Guide:

- Guide1Manager: Gestiona, en base a si el jugador ha alcanzado unos hitos, la guía número 1.
- Guide2Manager: Gestiona, en base a si el jugador ha alcanzado unos hitos, la guía número 2.
- Guide3Manager: Gestiona, en base a si el jugador ha alcanzado unos hitos, la guía número 3.
- Guide4Manager: Gestiona, en base a si el jugador ha alcanzado unos hitos, la guía número 4.

Intro:

- IntroManager: Administra las distintas animaciones y efectos del vídeo de introducción.

Island:

- bridgeManager: Gestiona la construcción de los puentes que conectan las islas.

Crafting:

- CampfireManager: Administra el encendido de la hoguera durante la noche.
- CraftingObject: Controla el slot dentro del menú de crafting para la creación de la hoguera.
- CraftingObject2: Controla el slot dentro del menú de crafting para la creación de la cama.
- CraftingShadow: La sombra de la construcción, previo a su completa construcción, se gestiona mediante esta clase.
- CraftingSlot: Hace un recuento de los materiales que tiene el jugador en el inventario para indicar visualmente si tiene las unidades necesarias o no.

HUD:

- InventoryInfoHUD: En la parte superior izquierda se muestran los diferentes objetos que va recogiendo el jugador. Esta clase es la encargada de la detección del objeto recogido y mostrar el texto correcto.
- InventorySlot: Gestiona cada slot que el jugador ve en el menú de inventario con las diferentes imágenes y las unidades de cada recurso.
- InventoryUI: Actualiza y administra el menú de inventario.
- InventoryQuickUI: Actualiza y administra el menú de inventario del acceso rápido
- StatsHUDManager: Es la clase encargada de actualizar visualmente las estadísticas del HUD, en base a las estadísticas del jugador.

- ToolBeltHUDManager: Gestiona la parte del HUD que referencia al cinturón de herramientas.
- StatHUD: Cada estadística tiene dos imágenes, una para cuando la ha perdido el jugador y otra para cuando el jugador tiene la estadística en positivo.
- ToolHUD: Clase que indica al jugador a través del HUD, si se está usando un slot de herramientas o no.

Recursos:

- AppleResource: Gestiona el recurso de manzana, una vez madura y cae al suelo..
- BerryFruitResource: Clase encargada de la gestión de la baya, una vez la ha recogido el jugador, iniciar el proceso de regeneración individual.
- BerryResource: El arbusto de bayas es gestionado mediante esta clase, que administra y controla la recolección de cada una de las bayas.
- MushroomResource: Gestiona el recurso setas.
- StoneOreResource: Clase encargada de revisar la resistencia de las menas de piedras e ir controlando y mostrando su destrucción progresiva.
- ToolResource: Clase que gestiona la obtención de las distintas herramientas iniciales.
- WoodAppleResource: Administra la maduración y caída de las manzanas del árbol frutal.
- WoodResource: El recurso obtenido de los diferentes árboles del juego, la madera, es gestionado con esta clase.
- RegenerationBerryResource: Clase que administra la regeneración del arbusto de bayas.
- RegenerationIndividualAppleResource: Clase que administra la regeneración individual de cada baya en el arbusto.
- RegenerationIndividualBerryResource: Clase que administra la regeneración individual de cada manzana del manzanero.
- RegenerationStoneResource: Gestiona la regeneración de las menas de piedra.
- RegenerationWoodAppleResource: Gestiona la regeneración de los árboles frutales.
- RegenerationWoodResource: Gestiona la regeneración de los árboles.

Character:

- AnimatorCharacterManager: Permite al controlador de las animaciones del personaje, gestionar correctamente el sistema de acciones.
- InventoryManager: Clase que alberga el inventario del jugador en un diccionario dinámico y todas sus funciones.
- PlayerEffectsManager: Administra los distintos efectos que puede causar el jugador, visuales y sonoros.
- PlayerManager: Clase que se encarga de gestionar todo lo relacionado con el jugador, estadísticas, acciones, estados, etc...

Animals:

- HorseManager: Clase que gestiona la IA de los caballos, el modo de patrulla y sus reacciones ante el jugador.
- LlamaManager: Clase que gestiona la IA de las llamas, el modo de patrulla y sus reacciones ante el jugador.
- IEnemySnakeState: Interfaz que implementan los distintos estados de la serpiente.

- enemyAISnake: Clase base que gestiona los distintos estados de la serpiente y sus reacciones.
- PatrolSnakeState: Administra el estado de patrulla de la serpiente y sus transiciones.
- AlertSnakeState: Administra el estado de alerta de la serpiente y sus transiciones.
- AttackSnakeState: Administra el estado de ataque de la serpiente y sus transiciones.
- IEnemySpiderState: Interfaz que implementan los distintos estados de la araña.
- enemyAISpider: Clase base que gestiona los distintos estados de la araña y sus reacciones.
- PatrolSpiderState: Administra el estado de patrulla de la araña y sus transiciones.
- AlertSpiderState: Administra el estado de alerta de la araña y sus transiciones.
- AttackSpiderState: Administra el estado de ataque de la araña y sus transiciones.

5. Testing

Se han realizado dos tipos de testing para asegurarse de la calidad del producto a presentar:

- Testing IT: Ha constado de tests unitarios para cada elemento del juego, con todas sus mecánicas. Una vez todos los elementos han sido probados, se ha testado todo el juego completo. Esta forma de testing por partes, ha asegurado la calidad del juego y se ha repetido un número elevado de veces debido a detección de errores en las iteraciones de testing.
- Testing con usuarios: Se ha conseguido involucrar en el proceso de testing a usuarios, para obtener información de elementos o formas de juego, que desde la visión de desarrollo se desconoce o es muy difícil su detección.

Los testing con usuarios, se han organizado en cuatro fases:

Fase	Contenido	Objetivo
1	Isla inicial	Enviar el juego sin información a los usuarios, para ver el flujo de juego y posible información no-implícita
2	Isla inicial y segunda isla	Ofrecer las guías a los jugadores y más información, como controles y objetivo. De esta forma, conseguimos saber si las mecánicas básicas quedan claras y el jugador sabe avanzar en el juego.

2.5	Isla inicial y segunda isla	Mismo contenido que la fase 2, con múltiples errores corregidos.
3	Mundo completo	Se testea todo el juego completo.

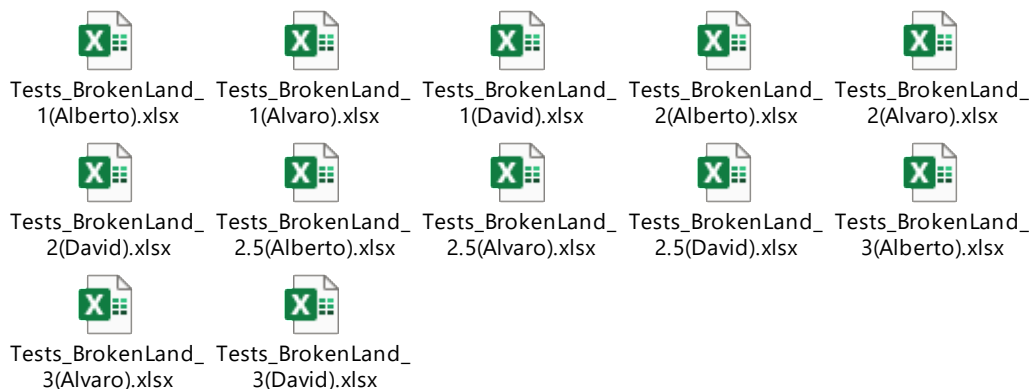
En cada fase de testing, se ha proporcionado tanto el ejecutable del juego, como un Excel a cumplimentar. El Excel informativo del testing contiene las diferentes áreas de tests, que han ayudado a la mejora del juego:

- Datos: Datos informativos del tipo de usuario y el tipo de configuración ha usado para jugar al juego.
- Flujo de juego: Un paso a paso, de lo que el usuario ha realizado dentro del juego. Ha servido para saber si había momentos en el que el jugador realizaba acciones no previstas, o se perdía por falta de información o recursos.
- Bugs: Errores encontrados al jugar.
- Mejoras: Respecto al contenido visto en las diferentes fases, posibles mejoras o cambios. No se han podido abarcar las mejoras propuestas, pero se han listado como mejoras futuras, para una versión posterior.
- Dudas: Uno de los aspectos más importantes que se ha aprendido a lo largo del desarrollo de BrokenLand, es que un juego puede despistar a un jugador si no le ofreces la información que necesita para jugar. Con este apartado se ha pretendido recopilar aquellas dudas, que el jugador tiene al jugar al juego.

Los usuarios que han participado en la fase testing y a los cuales agradezco que me hayan ayudado han sido:

- Alberto Martínez Rodríguez.
- Álvaro Martínez Rodríguez.
- David González Ingles.

Los diferentes excels obtenidos han sido:



6. Diseño de niveles

6.1 Mundo fragmentado

El mundo de BrokenLand se presenta como un único nivel, donde el jugador puede viajar libremente entre las islas mediante puentes.



Ilustración 127 Mundo de BrokenLand

El jugador inicia su partida en la llamada isla inicial, que es una isla ambientada en un denso bosque. Existen tres biomas bien diferenciados en el mundo de BrokenLand:

- **Bosque:** Islas en las que abunda la comida y sobre todo la madera. El jugador puede tener esta isla como base, debido a que no le faltará la comida.
- **Desierto:** Los recursos son escasos en este tipo de bioma y se encuentran los primeros enemigos del juego. No existe ninguna fuente de alimento en este tipo de islas.
- **Nieve:** Un bioma en el cual la nieve no para de caer. En este bioma el jugador puede encontrar algo de comida, pero sigue siendo escasa. Un bioma que aporta nuevos enemigos al juego.

BrokenLand se ha diseñado con el concepto de mundo fragmentado, para aportar un grado de novedad al género *survival*.

6.2 Isla inicial

El bioma de la isla inicial es el de bosque y representa la seguridad. El objetivo de esta isla es la introducción del jugador al mundo de BrokenLand, enseñándole lo básico para sobrevivir.

De esta forma, la tonalidad de la primera isla y el ambiente, es totalmente pacífico para que el jugador entienda que está ante un juego de supervivencia pacífico, para nada extremo.

La primera misión del juego es que el jugador deberá encontrar las tres herramientas escondidas a lo largo de la primera isla, para poder obtener recursos que le ayudarán a sobrevivir.



Ilustración 128 Mapa Isla Inicial 1

Una vez el jugador ya ha conseguido las tres herramientas, el jugador puede iniciar la recolección de materiales con el objetivo de sobrevivir y construir los diferentes puentes que se irá encontrando en su aventura.

Más allá de la infinidad de árboles que proporcionan madera en la primera isla, el jugador puede encontrar puntos específicos de la isla que le pueden proporcionar recursos valiosos o diferentes. Entre los que destacan los dos árboles frutales del juego y las menas de piedra.

Otro de los objetivos del diseño de esta primera isla es incitar al jugador a querer explorar y para conseguirlo, se añade un pequeño montículo, desde donde el jugador puede vislumbrar el resto de islas.



Ilustración 129 Mapa Isla Inicial 2

Con el objetivo de que el jugador salga preparado de esta segunda isla, se ha añadido una colina llena de setas, para que el jugador las coja, pruebe a comerlas y aprenda que es una fuente de alimentación válida dentro del juego

6.3 Segunda isla

La segunda isla se ha diseñado de tal forma que sea una introducción a lo que el jugador se encontrará en el futuro. Enemigos débiles y pocos recursos, es lo que define la segunda isla del mundo de BrokenLand.

En la segunda isla el jugador ya sabe qué hacer, y debe busca el siguiente puente para llegar a la última isla. De esta forma, los objetivos son claros, únicamente debe explorar y lidiar con los primeros enemigos.

Los enemigos de esta segunda isla, son los más débiles del juego, tanto en ataque como en vida, pero sirven de introducción para que el jugador entienda cómo funcionan los enemigos y sepa cómo enfrentarse a ellos.

Adicionalmente, uno de los objetivos principales de esta segunda isla es encontrar piel para poder fabricar una cama. La piel se puede obtener de los animales pasivos, en este caso las llamas. Por lo tanto, existe un objetivo secundario al llegar a esta isla y es la construcción de la cama, para poder dormir.

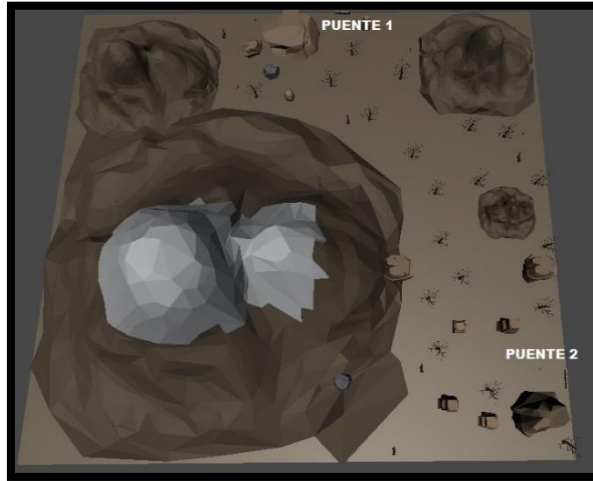


Ilustración 130 Mapa Segunda Isla 1

La isla presenta una montaña enorme, con el objetivo de que pueda ser vista desde cualquier punto del mundo de BrokenLand. Esta montaña es un símbolo que se ve desde la isla inicial y transmite la sensación de que hay mundo más allá de la primera isla; actúa como reclamo al jugador a querer continuar explorando.

En la segunda isla, los recursos escasean, debido al tipo de bioma que es, y el jugador debe llegar a ella con mucha comida o puede tener algún apuro y no salir vivo de ella mientras busca el siguiente puente.

Se han incluido dos menas de piedra, para evitar que el jugador se frustre al buscar los recursos necesarios para la construcción del segundo puente.

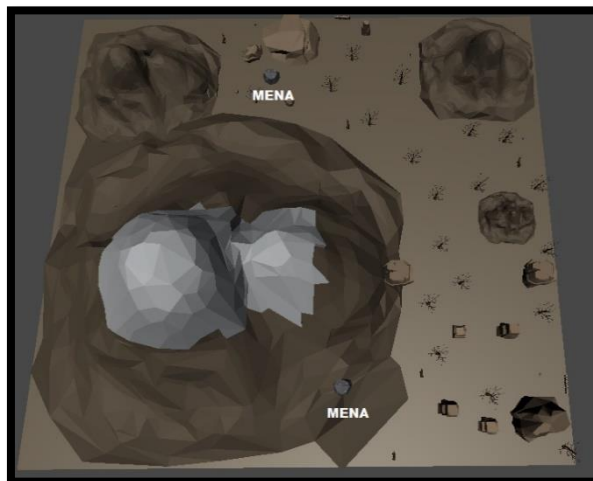


Ilustración 131 Mapa Segunda Isla 2

En la segunda isla, el jugador puede encontrarse con los siguientes animales:

Animal	Vida	Ataque
Llama	1	0
Serpiente	1	1

6.4 Isla destino

La creación de la isla final del juego, se ha diseñado con el objetivo de dejar al jugador con ganas de más. Se han incluido dos elementos que el jugador no espera, con tal de tener un aura de misterio al final del juego:

- Estructuras vikingas: Al parecer esta isla no es de esta época y parece que alguien ha vivido aquí recientemente. ¿Estamos en la línea temporal donde apareció el flash y fragmentó el mundo?
- Criaturas nunca antes vista. En la isla destino, el jugador puede encontrarse con arañas gigantes; algo que no existe en el mundo actual. De esta forma hace que el jugador empiece a plantearse que no está en el mundo actual y que ha sido trasladado a otro mundo diferente, lleno de peligros...

En esta isla, en el bioma nevado, no para de nevar; dando un toque místico y diferente al resto de islas que se ha ido encontrando el jugador. La isla está repleta de objetos que evocan a un tiempo antiguo, de la era vikinga. El jugador puede encontrarse desde un campamento vikingo, a edificaciones en las montañas que parecen ser sacadas de otro tiempo.

El objetivo principal de esta isla es la finalización del juego, por lo tanto, se ha incorporado un sencillo sistema de plataformas que el jugador debe superar para poder llegar a la cima de la estructura de piedra, encontrar el campamento vikingo y poder llegar a la brecha para finalizar el juego.



Ilustración 132 Mapa Isla Destino 1

En la última isla, el jugador puede encontrar nuevamente una fuente de comida, no de muy alta calidad, pero que le hará más leve su búsqueda del final del juego. Los alimentos que pueden encontrarse en la nieve son las setas, que fueron introducidas al jugador en la colina de las setas, de la primera isla. Adicionalmente, el jugador puede encontrar varias menas de piedra, por si necesita construir una hoguera para iluminar alguna zona.

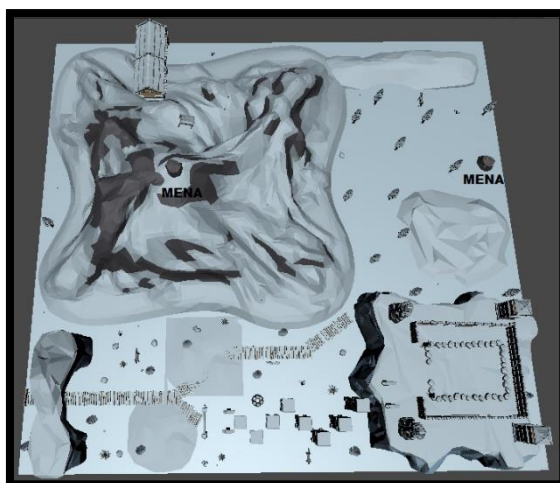


Ilustración 133 Mapa Isla Destino 2

En la tercera isla, el jugador puede encontrarse con los siguientes animales:

Animal	Vida	Ataque
Caballo	6	0
Araña gigante	4	2

7. Manual del usuario

7.1 Requerimientos técnicos

Hardware	Requerido/Probado
OS	Windows 7 o 10
API gráfica	DirectX 10,11 o 12
CPU	64 bits
RAM	8GB min.
HDD	200 MB
Resolución	1366x768 o 1920x1080
Input	Teclado y ratón

7.2 Controles

Manejo del personaje:

[W,A,S,D] Movimiento

[Shift] Correr

[Space] Saltar

[E] Tecla de acción, permite realizar acciones en base a la herramienta seleccionada.

Herramienta	Acción
Mano	Recoger objeto del suelo/ Recoger bayas/ Dormir en la cama
Hacha	Talar árbol

Pico	Picar mena de piedra
Cuchillo	Atacar a enemigos (el ataque es de dos corazones)

[Rueda del ratón] Cambiar de herramienta, en el cinturón de herramientas
 [Click en comida (en el inventario)] Comer
 [1,2,3,4] Comer

Menús:

[P / ESC] Abrir/Cerrar menú de pausa
 [I] Abrir/Cerrar menú de inventario
 [C] Abrir/Cerrar menú de crafting
 [C (Cerca de puente)] Iniciar construcción de puente
 7.3 Objetivos

El mundo ha cambiado después de un evento fatídico llamado El Flash. Como único superviviente tu misión será la de unificar el mundo que ahora aparece como fragmentado. Existen dos objetivos principales en BrokenLand:

- Sobrevivir: Busca recursos y comida, para sobrevivir en tu aventura.
- Busca la brecha: Una brecha ha aparecido en una isla lejana. ¿Qué será? El jugador debe encontrarla para averiguar más sobre lo que ha ocurrido.

7.4 Menús

Controles: Permite ver los controles, directamente desde el menú principal del juego.

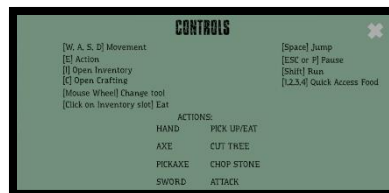


Ilustración 134 Menú de controles

Opciones: Desde el menú principal, se puede abrir el sub-menú de opciones, para poder configurar aspectos del juego, como dificultad, la opción de saltar la introducción o la opción de activar las guías o desactivarlas.



Ilustración 135 Menú de dificultad

Pausa: El menú de pausa sirve para parar el juego. Adicionalmente, el jugador puede volver al menú principal o salir del juego.



Ilustración 136 Menú de pausa

Inventario: En el menú de inventario, el jugador puede ver que recursos tiene encima. Además, el inventario sirve para comer, el jugador puede abrir el menú y hacer click encima de cualquier comida para comerla.



Ilustración 137 Menú de inventario

Crafting: El menú de crafting sirve para realizar construcciones que ayudan al jugador en su aventura.



Ilustración 138 Menú de crafting

7.5 Enemigos

Enemigo	Isla	Vida	Ataque
Llama	2	1	0
Serpiente	2	1	1
Caballo	3	6	0
Araña gigante	3	4	2

8. Trabajo futuro

A lo largo del desarrollo de BrokenLand, se han identificado una serie de mejoras y añadidos al juego, no planificados. Debido al tiempo limitado, no se han podido implementar, pero en futuras versiones se incluirán en BrokenLand.

Área de mejora	Mejora	Descripción
Apartado visual	Realizar nuevas animaciones para los animales	Para dotar de más realismo, se cambian las animaciones actuales, por unas realizadas desde cero.
Contenido	Más islas una vez se finaliza el juego	El juego continuará y se expandirá una vez el jugador llegue a la brecha temporal.
Contenido	Cultivos	El jugador podrá obtener semillas y cultivarlas para obtener comida.
Contenido	Más construcciones	Como, por ejemplo, casas o mobiliario.
Apartado visual	Posibilidad de cambiar el avatar	El jugador puede elegir el personaje inicial del juego, cada uno con características diferentes.
Contenido	Nuevas armas	Se incluirán armas de largo alcance (pica) y armas a distancia (arco).
Apartado visual	Mejoras en la cámara	Diferentes formas de utilizar la cámara, como, por ejemplo, cámara fija o moviéndola con teclado.
Contenido	Minimapa	Un minimapa que puede consultar el jugador para saber dónde está.

9. Conclusiones

La lección más importante aprendida a lo largo del desarrollo del trabajo es la importancia de dedicarle tiempo al diseño del videojuego. Muchas veces, la gente que dedicamos muchas horas a jugar a videojuegos, pensamos que todo es ponerse y crear contenido para el jugador.

Esto no es así, se necesita mucho tiempo, casi el mismo que el tiempo de desarrollo, para poder diseñar las mecánicas de un juego y que tengan una cohesión y que no parezcan partes que no tienen nada que ver, juntadas porque sí. Un caso práctico dentro de BrokenLand, son las construcciones, que se hicieron pensando muy bien que mecánicas introducir para que tenga sentido. Por el contrario, se tuvieron que rehacer de nuevos los recursos del juego, debido a que con la parte desarrollada no tenía sentido la inclusión de la regeneración de los recursos.

Si todos los elementos del juego se diseñan con tiempo y con cierta previsión de todas las mecánicas necesarias a incluir en el elemento, el desarrollo es único y no se debe volver a desarrollar ninguna parte del elemento. Esta forma de desarrollo, quizá no es tan importante en otros tipos de desarrollo, pero en desarrollo de videojuegos se ha demostrado (después de realizar el trabajo final de máster) que es vital para poder crear un videojuego.

Aproximadamente a la mitad del desarrollo de BrokenLand, se ha demostrado que la fase de testing no estaba bien planteada al inicio. Por propia experiencia, he aprendido que en cualquier tipo de desarrollo de software el apartado de testing es muy importante, pero sí que es cierto que, en otros tipos de desarrollos, este apartado se puede realizar al final de la fase de desarrollo y no impactar. En desarrollo de videojuegos, la fase de testing debe ser constante, debido a que tienes que jugar mucho al juego que estas desarrollando para encontrar posibles problemas o mejoras. En el diseño inicial de BrokenLand, la fase de testing estaba planificada al final, pero se ha tenido que ir abordando desde el inicio; en resumen, el testing debe acompañar al desarrollo.

Después de cuatro fases de testing con usuarios, me he dado cuenta que por mucho que (cómo desarrollador) pruebes el juego que estás desarrollando, nunca tendrás el mismo flujo de juego (a no ser que sea un juego lineal) que otros jugadores. Como desarrollador de videojuego, es imposible tomar las mismas decisiones y realizar las mismas acciones, que los jugadores que juegan a tu juego.

Estoy satisfecho con el trabajo realizado, debido a que se han cumplido todos los objetivos planteados al inicio de este trabajo de fin de máster. Si bien es cierto, que siempre hay margen de mejora, todas las mecánicas que estaban plasmadas en el diseño del juego, se han desarrollado. Consiguiendo un juego divertido, pacífico (un punto muy importante comentado en el diseño inicial) y haciendo que el jugador no tenga que pensar mucho para conseguir los objetivos. En los tiempos que corren, la mayoría de juegos son excesivamente complicados y en mi opinión, lo que los jugadores necesitamos a veces es desconectar y jugar a algo que no requiera pensar mucho.

Más allá de los objetivos planteados, se han realizado desarrollos adicionales de elementos que en un inicio no tenían importancia, pero que han cambiado por completo la forma en que se ve el juego, como, por ejemplo, el ciclo día y noche.

Estos desarrollos adicionales, han permitido de dotar al juego de más elementos que aportan variedad al juego y nuevas mecánicas, más allá de solo aportar más contenido. El resultado es un juego con más contenido que el previsto, con una duración adecuada y con todos los elementos del juego creados de forma que tengan coherencia dentro del mundo de BrokenLand; para evitar confundir al jugador.

Una parte de mi trabajo que me hubiera gustado mejorar es el diseño gráfico. Por falta de tiempo y conocimientos, no se ha podido diseñar ninguno de los modelos 3D del juego (más allá de alguna modificación sencilla o añadido). En mi opinión, si hubiera creado yo mismo los modelos, no hubiera elementos que quizá no encajan del todo (aunque se disimulan) como por ejemplo la cama, que no es realmente un modelo de una cama. Al crear uno mismo los modelos, sabe lo que tiene en mente y puede plasmarlo, de la otra forma, te tienes que adaptar a los modelos ya creados o retocar modelos existentes.

El testing es uno de los puntos que más ha afectado al desarrollo de BrokenLand, al ser un juego de mundo abierto, hay multitud de posibilidades en que el jugador puede realizar las acciones. Esto ha ocasionado que las primeras versiones del juego tuvieran multitud de errores y no se pudieran disfrutar como se preveía. De cara a próximos desarrollos, ha quedado patente que el testing se debe realizar desde el inicio, para que todas las versiones jugables creadas tengan una baja tasa de errores.

La planificación inicial del proyecto se ha seguido en mayor o menor medida, teniendo los objetivos claros de cada semana. La fase de testing ha tenido una duración muy superior a la esperada y eso ha ocasionado que se dediquen más horas de las previstas al desarrollo del juego, para corrección de errores o retoque de elementos. El tener que estar probando constantemente el juego y desarrollando a la vez, es algo que ha impactado en la planificación inicial; no causando ningún cambio en cuanto a los hitos marcados, sino en cuanto a horas dedicadas. El haber desglosado el proyecto en tareas repartidas a lo largo de las semanas, ha dejado claro desde el primer momento, que tareas abordar cada semana. De esta forma, he podido seguir en gran medida la planificación y cumplir los objetivos esperados de cada entrega.

Para abordar el desarrollo, se ha optado por una metodología incremental, dado que, para el tipo de juego, se ha decidido que era la adecuada. En BrokenLand cada isla es una entidad en sí, con sus elementos, de esta forma, desarrollando por completo todos los elementos de una isla antes de iniciar el desarrollo de los elementos de la siguiente; ha permitido desarrollar por separado cada isla prácticamente. De esta forma conseguimos que, una vez desarrolladas las mecánicas básicas del juego, el mismo va creciendo con nuevas mecánicas y de igual forma, el desarrollador debe incrementar el alcance de los tests a realizar. En mi opinión, para el género *survival* la metodología utilizada es la adecuada.

La siguiente etapa de BrokenLand está actualmente en diseño, reuniendo ideas y conceptos, que se plasmarán en una siguiente versión del juego. La idea de la siguiente versión de BrokenLand se basa en dos puntos importantes, progresión del jugador y más contenido.

La progresión del jugador es algo que no se ha podido explorar en la versión actual del juego por estar fuera del alcance y por tiempo; y desde el punto de

vista de un jugador de juegos del género de survival, no puede faltar. En una futura versión el jugador podrá mejorar sus armas y podrá utilizar armaduras; de esta forma se añaden nuevos elementos a construir por parte del jugador a modo de objetivos secundarios.

El concepto de mundo fragmentado se incluyó en BrokenLand como elemento diferenciador de otros juegos del mismo género; y se va a explotar de diversas maneras en la siguiente versión del juego. Se incorporarán más islas de los biomas ya creados (bosque, desierto y nieve), pero con distintas configuraciones. Adicionalmente, se incorporarán dos biomas nuevos. El primer bioma que está actualmente en diseño es el de Far West, un lugar con edificios que recuerda a las películas de vaqueros antiguas. Aquí el jugador se encontrará con humanos por primera vez, desde el Flash en el mundo. Más allá de la incorporación de humanos al juego, en este bioma el jugador podrá obtener un arma de largo alcance.

Uno de los biomas que más tiempo llevará de diseñar y de desarrollar es el de Jardín Japonés. Un lugar con muchos bosques de bambús, osos panda y ríos. En este bioma, el jugador encontrará nuevas formas de sobrevivir, con nuevas construcciones y nuevos tipos de armas, como las lanzas.

Desde mi punto de vista, queda muy claro hacia donde se dirige BrokenLand y el secreto detrás del Flash que asola la tierra en el juego; de momento me lo reservo como un secreto para mí dado que aún queda mucho por pensar y crear, hasta que BrokenLand esté completado y poder lanzarlo al mercado.

10. Glosario

Término	Explicación
Survival	Género de videojuegos que engloba a juegos de supervivencia, donde el jugador debe sobrevivir.
Bioma	Entorno que identifica un espacio de un juego de supervivencia. Por ejemplo, bosque, nieve, etc...
Extreme-Survival	Evolución del género survival, que se enfoca en ofrecer al jugador un entorno muy hostil, siendo de alta dificultad su entendimiento y manejo
Mundo Fragmentado	Mundo de juego en el cual no es posible viajar entre todos los puntos del mapa, sino que se tiene que unificar el mundo para poder acceder a todas las partes
Brecha temporal	Efecto eléctrico, similar a los portales de las películas de ciencia ficción, en donde finaliza el juego de BrokenLand.
Flash	Evento que ocurrió en el mundo de BrokenLand y causó la fragmentación del mundo. Se desconoce su procedencia.

11. Bibliografía y Ludografía

11.1 Bibliografía

ARANDA, Daniel; GÓMEZ, Salvador; NAVARRO, Víctor; PLANELLS, Antonio José (2015) Game & play: diseño y análisis del juego, el jugador y el sistema lúdico. Barcelona: UOCPress

BRADLEY, Alan. Balancing survival gameplay and RPG progression in Conan Exiles (2017) [en línea] USA: GAMASUTRA [Consulta: 24 de febrero de 2020] <https://www.gamasutra.com/view/news/292110/Balancing_survival_gameplay_and_RPG_progression_in_Conan_Exiles.php>

DUCH, Jordi; TEJEDOR, Heliodoro; KANAAN, Samir; ESCUDERO, Gerard (2019) Inteligencia Artificial para videojuegos Barcelona: UOC

FRANCIS, Bryant. The difference between making an MMO versus an Early Access survival game (2017) [en línea] USA: GAMASUTRA [Consulta: 19 de febrero de 2020] <https://www.gamasutra.com/view/news/296397/The_difference_between_making_an_MMO_versus_an_Early_Access_survival_game.php>

GONZÁLEZ, Rafael; BOURDIN, Pierre (2019) First Person Shoter en Unity 3D Barcelona: UOC

GONZÁLEZ, Rafael; BOURDIN, Pierre (2019) Un juego de plataformas 3D en Unity 3D Barcelona: UOC

VINCENT, Brittany. The 25 Best Survival Games (2019) [en línea] USA: POPULAR MECHANICS [Consulta: 10 de febrero de 2020] <<https://www.popularmechanics.com/culture/gaming/g30223018/best-survival-games/>>

11.2 Ludografía

Minecraft (Mojang Studios, 2009)

The Long Dark (Hinterland Studio, 2017)

Conan Exiles (Funcom, 2018)

YLands (Bohemia Interactive, 2019)