



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Neyteria, un juego 2D de plataformas. Aspectos técnicos y gestión.

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Pousa Barros, Gianfranco

Tutor/a: Abad Cerdá, Francisco José

CURSO ACADÉMICO: 2021/2022

RESUMEN

Este TFG propone el diseño y la implementación del videojuego de plataformas, acción y aventuras llamado "Neyteria". El personaje principal será un humano medio ciborg, modificado con tecnología avanzada, con un brazo robótico que le permite, no solo tener una fuerza sobrehumana, sino también rápidos reflejos a la hora de combatir. La estética del juego será Pixel Art, con distintos tonos de color para cada nivel. Se trata de un juego offline y de un solo jugador, con dos modos de juego: normal y difícil. Los distintos niveles serán finitos, y su dificultad irá aumentando de forma que los enemigos de niveles más avanzados tendrán más nivel y no dejarán tantas recompensas. Para finalizar el juego, se deberá avanzar a través de distintos niveles saltando, golpeando, corriendo, esquivando y usando habilidades y artefactos, para enfrentarse a los distintos enemigos y matar al jefe de cada nivel hasta pasarse el nivel final. El juego contará con más de 10 enemigos, con distintas características y habilidades.

Para el desarrollo del proyecto se utilizará Unity, aplicando principalmente los conocimientos obtenidos en la Asignatura Optativa de cuarto año "Desarrollo de Videojuegos 2D", y haciendo uso también del resto de conocimientos adquiridos a lo largo del Grado de Ingeniería Informática. Se seguirá una metodología de desarrollo ágil, la cual permite entregar de forma frecuente sistemas de software diseñados con iteraciones rápidas.

Este TFG se desarrollará por un equipo de dos estudiantes. Gianfranco Pousa, autor de esta memoria, se encargará principalmente de los aspectos más técnicos, pero también a la gestión y planificación del juego. De una manera más específica, se encargará de: diseño de niveles, objetos y principales mecánicas del juego, movimiento del personaje, sistemas de vida, estamina y animaciones principales del protagonista, código relativo a los módulos y habilidades del personaje, desarrollo de la inteligencia de los enemigos y de la realización de los jefes, diseño de la estética del juego y de la trama principal e historia del mismo, planificación de las diferentes versiones y entregables, diseño de un sistema funcional de inventario, y gestión de carga de niveles.

Palabras clave: videojuego, plataformas, 2D, Unity, Neyteria.

ABSTRACT

This TFG proposes the design and implementation of the platform, action and adventure video game called "Neyteria". The main character will be a half-cyborg human, modified with advanced technology, with a robotic arm that allows him, not only to have superhuman strength, but also quick reflexes when fighting. The aesthetic of the game will be Pixel Art, with different shades of color for each level. It is an offline and single-player game, with two game modes: normal and difficult. The different levels will be finite, and their difficulty will increase so that the enemies of more advanced levels will have more level and will not leave so many rewards. To finish the game, you must advance through different levels by jumping, hitting, running, dodging and using skills and artifacts, to face the different enemies and kill the boss of each level until the final level is passed. The game will feature more than 10 enemies, with different characteristics and abilities.

For the development of the project Unity will be used, applying mainly the knowledge obtained in the Optional Subject of the fourth year "Development of 2D Video Games", and making use of the rest of the knowledge acquired throughout the Degree in Computer Engineering. An agile development methodology will be followed, which allows frequent delivery of software systems designed with rapid iterations.

This TFG will be developed by a team of two students. Gianfranco Pousa, author of this report, will be mainly responsible for the more technical aspects, but also for the management and planning of the game. In a more specific way, it will be responsible for: design of levels, objects and main mechanics of the game, movement of the character, life systems, stamina and main animations of the protagonist, code related to the modules and skills of the character, development of the intelligence of the enemies and the realization of the bosses, design of the aesthetics of the game and the main plot and history of the same, planning of the different versions and deliverables, design of a functional inventory system, and management of level loads.

Keywords: videogame, platforms, 2D, Unity, Neyteria.

INDICE

1. INTRODUCCION	4
2. ESTADO DEL ARTE	6
2.1. Super Mario Bros	8
2.2. Hollow Knight y Ori	10
2.3. Dark Souls.....	11
3. ANALISIS DE REQUISITOS	14
3.1. Características del juego.....	14
3.2. Herramientas técnicas	17
3.2.1. Unity.....	17
3.2.2. Visual Studio.....	17
3.2.3. Git y GitHub.....	18
3.2.4. HackNPlan	19
4. DISEÑO DE LA SOLUCION	21
4.1. Desarrollo del personaje.....	21
4.2. Desarrollo del equipamiento	23
4.3. Objetos	24
4.4. Animaciones.....	26
4.5. Menús.....	28
4.6. Enemigos.....	30
4.7. Niveles y optimización.....	32
4.8. Diálogos y mensajes.....	34
5. PRUEBAS Y TESTING	37
6. CONCLUSIONES	41
6.1. Relación del trabajo con los estudios cursados.	41
7. TRABAJOS FUTUROS	43
8. REFERENCIAS	44
9. ANEXO	46
9.1. Licencias.	46
9.2. Objetivos de Desarrollo Sostenible.	47
9.3. <i>Game Design Document</i>	47

1. INTRODUCCION

Actualmente la relevancia de los videojuegos es alta y, cada año que pasa, crece aún más. De hecho, el ámbito de los videojuegos se refleja cada vez más en actividades cotidianas o incluso en cualquier canal de televisión, ya que forman parte de las noticias actuales y de muchísimos eventos retransmitidos y seguidos por muchas personas y grandes plataformas de transmisión de videos o *streams*.

Pese a que en tres últimos años están creciendo cada vez más los videojuegos 3D de tipo mundo abierto, a muchísima gente le siguen interesando los videojuegos 2D de tipo plataforma, ya que también tienen el objetivo de explorar y desplazarse por un mapa con recorridos.

Este proyecto, presentado como Trabajo de Fin de Grado, consta del desarrollo de un videojuego de plataformas 2D donde el personaje principal explora y recorre los distintos niveles aprovechándose de ciertas habilidades para combatir o superar pruebas. Se ha decidido que el videojuego sea 2D principalmente por la sencillez frente a uno 3D y por el escaso tiempo de desarrollo, además de poseer los conocimientos únicamente sobre el desarrollo de un videojuego en 2D. Para el desarrollo del videojuego se ha utilizado el motor de Unity.

El proyecto ha sido desarrollado por dos estudiantes de cuarto curso del Grado en Ingeniería Informática, concretamente David Garrido Palacios y Gianfranco Pousa Barros, quienes se ha encargado de:

- David Garrido Palacios:
 - o La realización del menú principal, menú de pausa y pantalla de muerte
 - o Desarrollo del código de mecánicas básicas de combate y detalles del movimiento del personaje.
 - o Animación básica de las armas del protagonista.
 - o Diseño y elección de los enemigos del juego.
 - o Desarrollo de los enemigos del juego.
 - o Diseño y desarrollo de la animación de los enemigos.
 - o Diseño y desarrollo de los niveles del juego.
 - o Desarrollo e implementación del sistema de diálogo.

- Desarrollo de parte de la historia del juego.
 - Representación de la historia a lo largo del juego.
 - Desarrollo artístico del HUD del jugador.
 - Diseño e implementación del sistema de sonido y selección de estos.
 - Realización de elementos y eventos para puzzles en los niveles del juego.
 - Selección de los principales *sprites* colores y fuentes del juego acordes a la temática.
- Gianfranco Pousa Barros:
- Diseño y desarrollo de los objetos y armas del juego.
 - Diseño e implementación de las principales mecánicas del juego.
 - Diseño e implementación del movimiento del personaje.
 - Desarrollo de los sistemas de vida y estamina del protagonista.
 - Animaciones principales del protagonista.
 - Realización del código de las habilidades del personaje.
 - Desarrollo del sistema de módulos.
 - Diseño de la estética principal del juego.
 - Diseño de la trama principal e historia del juego
 - Planificación de las distintas versiones y entregables de este, con fechas acordadas.
 - Diseño y desarrollo del sistema de inventario.
 - Diseño del sistema de tienda del juego.
 - Diseño e implementación del sistema de estadísticas del protagonista.
 - Diseño e implementación del sistema de niveles de los enemigos.
 - Gestión de carga de niveles para optimización del juego.
 - Diseño y desarrollo del sistema y movimiento de la cámara dentro del juego.

2. ESTADO DEL ARTE

Los videojuegos 2D de género plataforma son uno de los géneros de videojuegos en dos dimensiones más conocidos, donde el personaje principal debe desplazarse por los distintos niveles saltando entre plataformas en el aire. Este tipo de juegos suelen tener distintas mecánicas, trampas y puzzles o acertijos a resolver para poder continuar explorando la zona, además de un sistema de ataque, ya sea a corta distancia o con disparos, para defenderse de enemigos o del entorno.

Uno de los primeros videojuegos 2D que se puede considerar de plataformas es *Donkey Kong*, desarrollado en el año 1981 por Nintendo. El único objetivo del juego era escalar por las plataformas inclinadas a la vez que se saltaban los barriles que lanzaba *Donkey Kong* desde arriba y que bajaban deslizándose por las rampas, como se puede observar en la Figura 1. Realmente no había nada de exploración, ya que eran niveles que no tenían por qué estar conectados. Más adelante llegó *Super Mario Bros*, juego de Nintendo desarrollado en 1985 que revolucionó la industria de los videojuegos. Este sí que tenía los aspectos principales de un videojuego de plataformas descritos en el párrafo anterior, con exploración de niveles, enemigos, acertijos, recolección de objetos, etc. *Super Mario Bros* provocó un gran cambio en el desarrollo de los videojuegos, ya que muchísimos videojuegos imitaron este estilo de juego. En la Figura 2 se puede ver a Mario rodeado de algunas plataformas y saltando por encima de un enemigo.

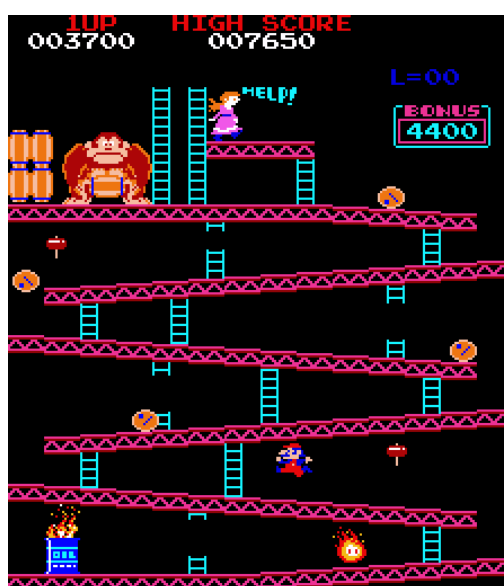


Figura 1. Videojuego *Donkey Kong* (1981)



Figura 2. Videojuego *Super Mario Bros* (1985)

Otros videojuegos que se han utilizado como referentes, de la actualidad y conocidos, serían juegos como *Hollow Knight* (2017) o como las dos entregas *Ori and the Blind Forest* (2015) y *Ori and the Will of Wisps* (2017). Estos, a diferencia del comentado *Super Mario Bros*, destacan por tener un diseño gráfico muy desarrollado, con efectos visuales destacables y otras mecánicas que permiten fortalecer al personaje principal a medida que avanza en el juego, como un sistema de habilidades que se puede ir desbloqueando. Además, con esas habilidades obtenidas, se podrán explorar ciertas zonas que exigen esa habilidad para ser descubiertas, o incluso zonas por las que ya se haya pasado. A estos juegos se le atribuyen el género conocido como “Metroidvania”, que adopta el nombre de los juegos *Metroid* y *Castlevania*. Los juegos con este género se caracterizan por tener una exploración en un mundo conectado donde algunas zonas no pueden ser exploradas al llegar a ellas, sino más adelante después de haber obtenido ciertos objetos o habilidades. En las Figuras 3 y 4 se pueden ver ejemplos del juego de *Hollow Knight* y *Ori an the Will of Wisps* respectivamente.

Una mecánica que se ha implementado en nuestro videojuego está inspirada en los juegos tipo “souls-like”. Este término deriva del videojuego *Dark Souls*, y se atribuye a aquellos juegos que tienen mecánicas similares a las de este juego, como la gran dificultad que ofrecen al jugador, el sistema de mejora de los atributos y el escalado de estos, la penalización al momento de morir, etc. En este caso, se trata de la mejora de algunos de los atributos del personaje a cambio de algún recurso del juego y otras mecánicas como el gasto de resistencia al atacar en función del peso.



Figura 3. Videojuego *Hollow Knight* (2017)



Figura 4. Videojuego *Ori and the Will of Wisps* (2017)

2.1. Super Mario Bros

El videojuego *Super Mario Bros* impulsó a muchos otros juegos a adoptar el tipo de género plataforma e incorporar las mecánicas de recorrer un nivel con enemigos que esquivar y matar a la vez que saltas por el terreno entre las distintas plataformas. El juego tuvo tanto éxito que, incluso hoy en día, se siguen desarrollando juegos de la saga *Mario Bros* de tipo plataforma con el mismo estilo que aquel juego del año 1981, pero gráficamente mejorados y con mecánicas y niveles innovadoras.

El objetivo principal del juego *Super Mario Bros* es superar todos los niveles para poder rescatar a la Princesa Peach de las garras del malvado Bowser. Cada nivel consta de una mezcla de plataformas que forman el terreno del nivel, ya sean el propio suelo fijo, plataformas flotantes o en movimiento, y otros tipos de plataformas con mecánicas distintas. En juegos posteriores de la saga *Super Mario Bros*, se incorporan plataformas con funciones especiales como los champiñones gigantes en los que el personaje sale propulsado hacia arriba.

Otra característica importante en este juego son los distintos *power-ups*, objetos que, al tocarlos con el personaje, le proporcionarían distintos tipos de ventajas o desventajas, como lanzar bolas de fuego y de hielo, hacerse pequeño o gigante, o incluso ser inmortal y matar a cualquier enemigo con solo tocarlo. Se suelen obtener al saltar debajo de los bloques amarillos con interrogación. El uso de estos objetos no son solo una manera de facilitar el juego, sino en algunos casos imprescindibles para poder acceder a ciertas partes de los niveles o para poder resolver algunos acertijos. En la Figura 5 se puede ver a Mario entre plataformas con un *power-up* y bloques de objetos con el símbolo de interrogación.



Figura 5. Escenario de *Super Mario Bros* con bloques “?” y un *power-up* champiñón.

2.2. Hollow Knight y Ori

Hollow Knight, junto a *Ori and the Blind Forest* y *Ori and the Will of Wisps* son videojuegos muy conocidos y jugados en la actualidad a pesar de que la mayoría de los éxitos de estos últimos años sean videojuegos 3D. Una de las características que hacen esto posible es su gran calidad gráfica, pero por otro lado destacan en ciertos aspectos que se describen a continuación.

Los videojuegos como *Hollow Knight* y *Ori* están considerados como género “Metroidvania”. Este tipo de juegos incorporan la mecánica de mejorar las capacidades del personaje principal, no solo con objetos que proporcionan un *power-up* temporal al personaje, sino a través de habilidades que una vez se aprenden y desbloquean, ya no se pueden olvidar y permanecen durante el resto del juego. Esta mecánica no solo sirve para hacer más ameno el juego, sino para poder explorar zonas de otros niveles que precisen de esa habilidad para descubrirlas o completarlas. De hecho, el término “Metroidvania” hace referencia a un juego de plataformas donde la exploración no es totalmente lineal, sino que, habiendo avanzado en el juego, se puede volver a zonas anteriores para explorar y descubrir nuevas áreas con las habilidades obtenidas durante el progreso.

En algunos casos se pueden obtener mejoras del personaje que se pueden aplicar o quitar a gusto del jugador. En el caso de *Hollow Knight*, durante el progreso del juego se pueden obtener unos Amuletos que el personaje se puede equipar. Cada Amuleto le proporciona un *power-up* distinto y cuestan de equipar una determinada cantidad de muescas, como se observa en la Figura 7. La cantidad máxima de muescas se va incrementando con el progreso del juego. En el caso de *Ori*, puedes canjear puntos de energía a cambio de habilidades, como se ve en la Figura 6.



Figura 6. Árbol de habilidades desbloqueables del juego *Ori and the Blind Forest*.



Figura 7. Selección de Amuletos del juego *Hollow Knight*.

2.3. Dark Souls

Dark Souls es otro juego con mucha reputación que, junto a los siguientes juegos de la saga desarrollados por FromSoftware, son reconocidos por su gran dificultad. De estos juegos de exploración, acción y mazmorras, surge el término y género *souls-like*, que se les atribuyen a todos aquellos videojuegos que comparten algunas mecánicas características de la saga *Souls* o que se inspiran en ellas.

Una de las características típicas de los *Souls* es la de poder aumentar las estadísticas del protagonista a cambio de cierto recurso que sirve como moneda y que se puede obtener derrotando a los enemigos. En el caso de *Dark Souls* son las almas y, con ellas, se puede subir cada atributo de nivel (hasta un máximo de noventa y nueve por cada atributo). Cada nivel que se suba hará que el siguiente requiera de más almas que el anterior, y así sucesivamente. En “Neyteria” se ha adaptado esta mecánica, pero con muchos menos atributos. Se podrá elegir un nivel a subir entre cinco atributos que parten originalmente con nivel cero, pudiendo alcanzar cada uno el nivel diez. Es necesario subir esas estadísticas para que a medida que se avanza en el juego, sea posible derrotar a los enemigos o sobrevivir.

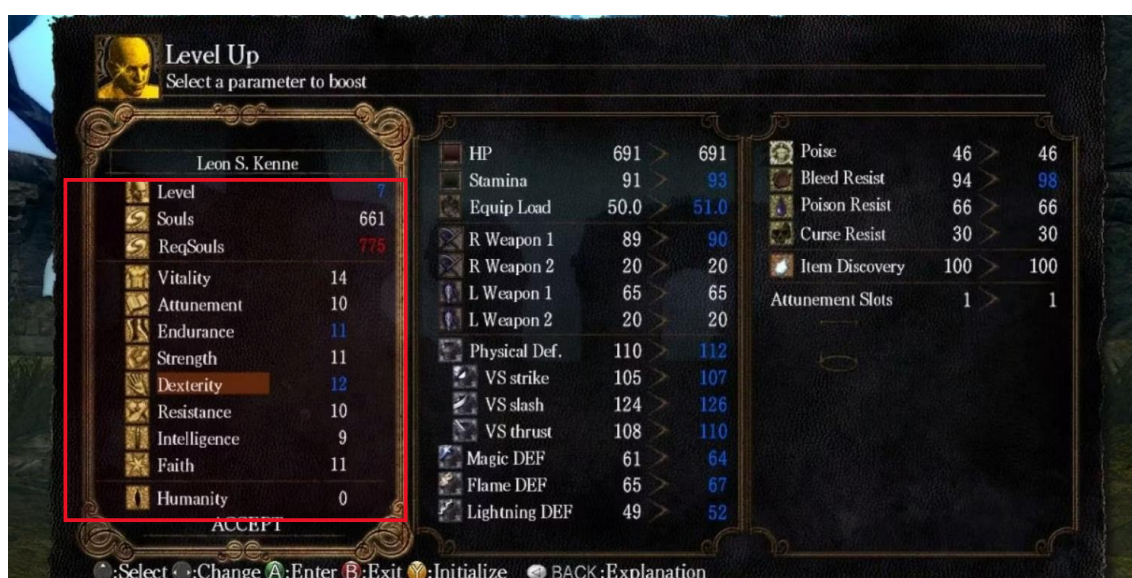


Figura 8. Menú de atributos de *Dark Souls*.

Otra característica muy común de la saga es la de que las armas y las piezas de armaduras que se equipe el personaje posean habilidades pasivas, aunque esta mecánica puede pertenecer perfectamente a muchos otros tipos de juegos. En el caso del videojuego desarrollado, los trajes con los que el jugador puede equipar al protagonista tienen habilidades pasivas que le benefician de alguna manera, ya sea directa o indirectamente, o a cambio de alguna desventaja como reducir las capacidades de otros atributos o aumentar la dificultad del entorno.

Una última característica que destaca en *Dark Souls* es la existencia del atributo “Carga” o “Peso” del personaje principal. Las armas y armaduras que se equipen tienen un determinado peso que

se termina sumando para hacer un peso total. En *Dark Souls* es imprescindible rodar cuando un enemigo ataque al jugador, pero la cantidad de peso con la que cargue el personaje influye en la capacidad y velocidad de rodar. Si se supera el peso máximo, que se muestra en la Figura 9, los ataques con las armas no harán daño prácticamente. Además, al atacar con el arma, se consumirá una cantidad de resistencia en función del peso. Así pues, cuanto más peso, más difícil es atacar y esquivar los ataques de los enemigos.

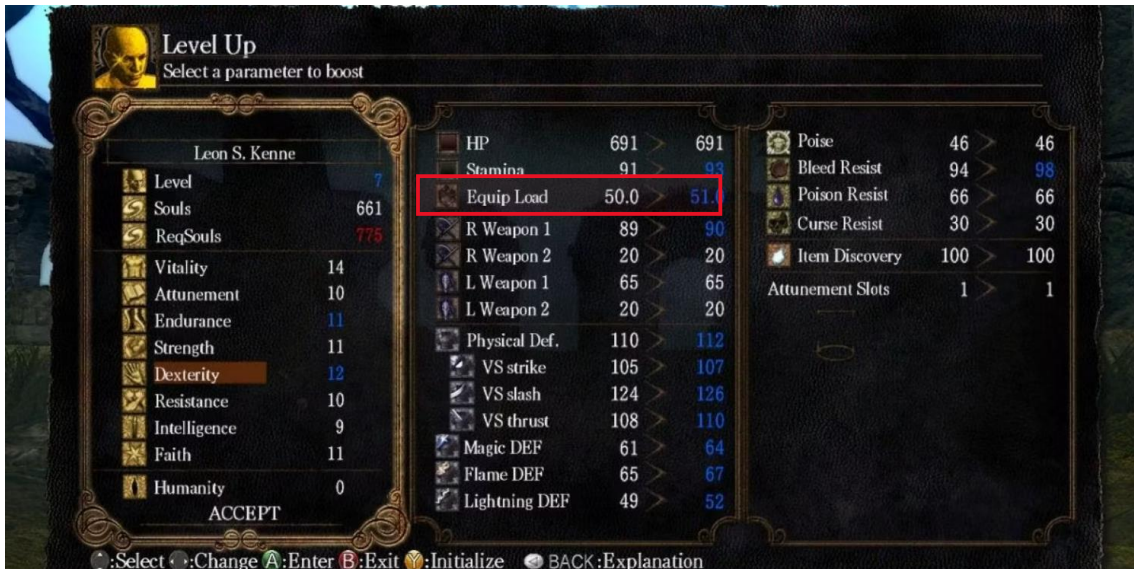


Figura 9. Carga de equipo máxima (peso) de *Dark Souls*.

COMPARATIVA DE CARACTERISTICAS ENTRE JUEGOS				
Juegos	<i>Super Mario Bros</i>	<i>Hollow Knight</i>	<i>Dark Souls</i>	<i>Neyteria</i>
Plataformas	NES	Nintendo Switch PlayStation 4 Xbox One PC	Nintendo Switch PlayStation 3/4 Xbox 360/One PC (Windows)	PC
Videjuego 2D	Sí	Sí	No	Sí
Género	Plataformas	Metroidvania	Mazmorras, Rol de acción	Plataformas Metroidvania
Multijugador	No	No	Sí	No
Personaje “customizable”	No	No	Sí	No
Mejoras del personaje	No	Sí	Sí	Sí

3. ANÁLISIS DE REQUISITOS

3.1. Características del juego

Cuando se propone realizar el desarrollo de un videojuego es muy importante tener claro, no solo qué tipo de videojuego se tiene en mente crear, sino también qué características se pretenden implementar.

La principal idea de “Neyteria” surge de la combinación de diversas características que los desarrolladores tenían pensadas uno o dos años atrás, pero aprovechando la oportunidad que se daba con la asignatura “Desarrollo de Videojuegos 2D” de la Escuela Técnica Superior de Ingeniería Informática en la Universidad Politécnica de Valencia, se pudo llevar a cabo este proyecto. Las características básicas que se plantearon eran que fuese un videojuego 2D de tipo plataformas donde existían varios niveles que estaban conectados entre sí, dando la sensación al jugador de que no hay pantallas de carga y de que es como un mundo abierto donde todas las zonas están unidas. Para moverse de un nivel a otro, solo se podría hacer desde una zona base que conecta todas las zonas. En la Figura 10 se muestra el mapa de los niveles conectados con los teletransportes disponibles.



Figura 10. Mapa de Comunicación del Nexo Central.

En cuanto al personaje principal, éste posee tres tipos de arma para defenderse: espada corta, ligera y con un daño normal; maza, más pesada y con gran daño; y cañón láser, poco daño y a larga distancia. Además, el personaje también contaría con la capacidad de equiparse un traje que le otorgue una habilidad, pudiendo elegir entre varios trajes distintos. Pero la característica especial del jugador es que tiene un brazo robótico donde pueden incrustarse distintos tipos de artefactos llamados “Módulos del Ocaso”.

Estos módulos tienen la capacidad de desencadenar el potencial oculto del brazo robótico, de manera que le proporciona habilidades especiales al personaje. Estas habilidades o efectos proporcionados tienen su tiempo de eficacia, durante el cual la habilidad está activada y permanece funcionando, y su tiempo de enfriamiento, para poder volverlas a activar. El brazo posee tres ranuras de módulos, pero algunos módulos necesitan de más de una ranura para funcionar, por lo que a veces no será posible equiparse tres módulos al mismo tiempo. La diferencia entre las habilidades de los trajes y de los “Módulos del Ocaso” es que las primeras son habilidades pasivas, es decir, que están siempre funcionando mientras se tenga equipado el traje, mientras que las de los módulos son activas y se podrán ejecutar cuando el jugador lo desee, siempre y cuando el módulo no se encuentre en periodo de enfriamiento.

Las bases desde las cuales uno puede transportarte a otras zonas se llaman “Nexo Central”. Si el personaje principal es derrotado, aparecerá en el último nexo central al que estuvo conectado. En estas mismas bases es donde se puede cambiar el equipamiento del personaje principal, además de acceder a otras funciones como una tienda de objetos y un menú para subir los niveles de los atributos del personaje. Tanto la subida de niveles de los atributos como la compra de los objetos en la tienda se realizarán principalmente a cambio de la moneda del juego conocida como “Degiterio”, la cual se puede obtener de diversas formas, entre ellas al derrotar enemigos del mundo o al abrir suministradores (equivalentes a cofres).

Esparcidos por todo el mundo, se encuentran los conocidos como suministradores, pedestales mediante los que se pueden obtener varios objetos si el personaje lo activa. En algunas ocasiones, estos suministradores pueden estar bloqueados por algún grupo de enemigos o por algún acertijo a resolver. De igual manera, hay zonas del mapa a las que solo se puede acceder si se completa cierto puzle o si se usa algún “Módulo del Ocaso” en específico. Al final de cada uno de los niveles, justo antes de pasar al “Nexo Central”, siempre habrá un jefe final al que habrá que derrotar para poder continuar avanzando.

A medida que se avanza en los niveles, el personaje principal se irá encontrando con diversos enemigos. Los enemigos del mundo están clasificados por su nivel y, la mayoría de sus atributos dependen de este nivel, de manera que escalan exponencialmente con este. Durante el tiempo de juego, existe un ciclo de día y noche. Cuando cae la noche, el nivel de los enemigos se incrementa en cinco y, por lo tanto, sus atributos también se ven afectados. Por otro lado, cuando es de día, hay una cierta probabilidad de que se desate una tormenta, haciendo que el nivel de los enemigos se reduzca en cinco. Además, cuando es de día y no hay tormenta, hay una pequeña probabilidad de que se produzca un Eclipse, que provocará que el nivel de todos los enemigos se incremente en un rango de seis a diez.

El mundo jugable del videojuego se distribuye en cuatro niveles. El primer nivel consta de un conjunto de grutas que se encuentran en las cavernas profundas del mundo, donde la luz solar no llega y se debe iluminar la zona con luz propia, aunque algunos objetos o minerales brillan en la oscuridad. El segundo nivel se acerca un poco más a la superficie, pero sigue permaneciendo bajo tierra, pues el terreno sigue siendo rocoso y subterráneo, pero con un poco más de vegetación y algo de luz solar, así que ya no es necesario llevar luz propia. El tercer nivel está formado por dos zonas: La primera se encuentra en un campo abierto, ya en la superficie, acompañado de un gran árbol que se puede escalar y ascender por él; la segunda zona está sumergida en un bosque muerto y termina con una catedral antigua desde la que se puede acceder al último nivel. El cuarto nivel se encuentra en lo más profundo del mundo, incluso por debajo del primero, predominando así grandes grutas de lava y feroces enemigos. En la Figura 11 se puede ver un boceto original de la distribución de los niveles y las zonas del juego.

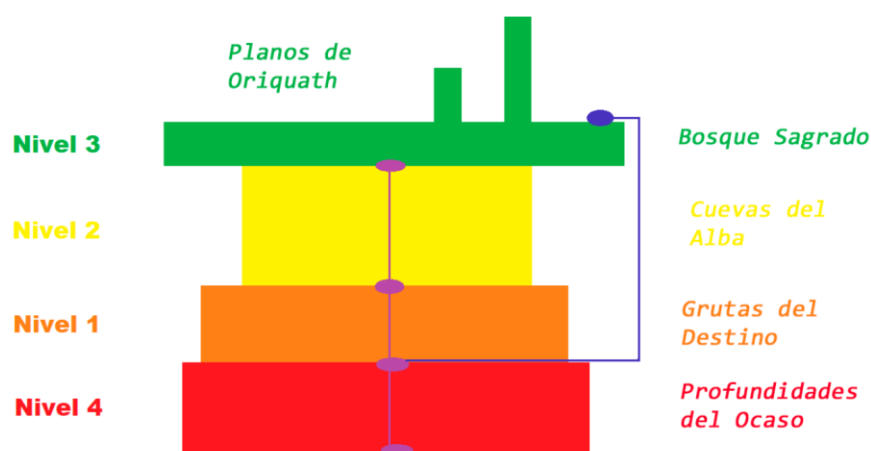


Figura 11. Boceto de los niveles unidos por la comunicación de los Nexos.

3.2. Herramientas técnicas

Es imprescindible para el correcto, cómodo y eficiente desarrollo de un juego el uso de buenas herramientas. En esta sección del documento se plantearán una serie de herramientas que permitirán la creación del videojuego 2D y la implementación de este.

3.2.1. Unity

Unity es uno de los motores gráficos para desarrollo de videojuegos más conocidos. Se ha optado por esta herramienta debido a su amplio repertorio de opciones que ofrece a la hora de diseñar escenas, mecánicas, animaciones y por la gran compatibilidad con diferentes programas y plataformas. Además, esta es la herramienta que se ha utilizado durante las sesiones de la asignatura “Desarrollo de Videojuegos 2D”, misma donde se comenzó el desarrollo de “Neyteria”. En la Figura 12 se puede observar la herramienta Unity, que proporciona un editor de escenas, con una jerarquía ordenada de los objetos de la escena y un inspector que visualiza todas las propiedades de un objeto seleccionado.

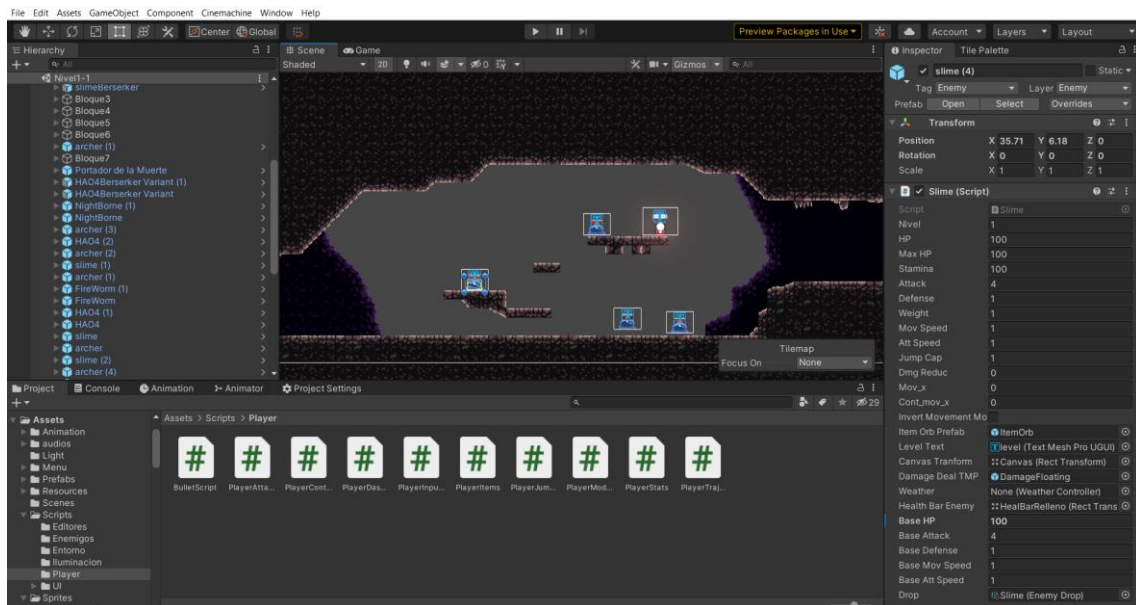


Figura 12. Proyecto “Neyteria” abierto con Unity.

3.2.2. Visual Studio

Una buena herramienta para programar y editar scripts a la hora de desarrollar un videojuego es Visual Studio. De hecho, esta ofrece una alta compatibilidad y sincronización con Unity, ya que proporciona documentación y sugerencias a la hora de escribir código y permite la depuración de

este, haciendo posible detener la ejecución de Unity en ciertas líneas de código. Esto facilita la detección y búsqueda de errores de programación. Visual Studio es una herramienta que se ha empleado a lo largo del grado de Ingeniería Informática y en la asignatura de “Desarrollo de Videojuegos 2D”. En la Figura 13 se observa la aplicación Visual Studio con parte del código de un script en el lenguaje de programación C# y con algunos “puntos de ruptura” o *breakpoints* en los que la ejecución se detendrá.

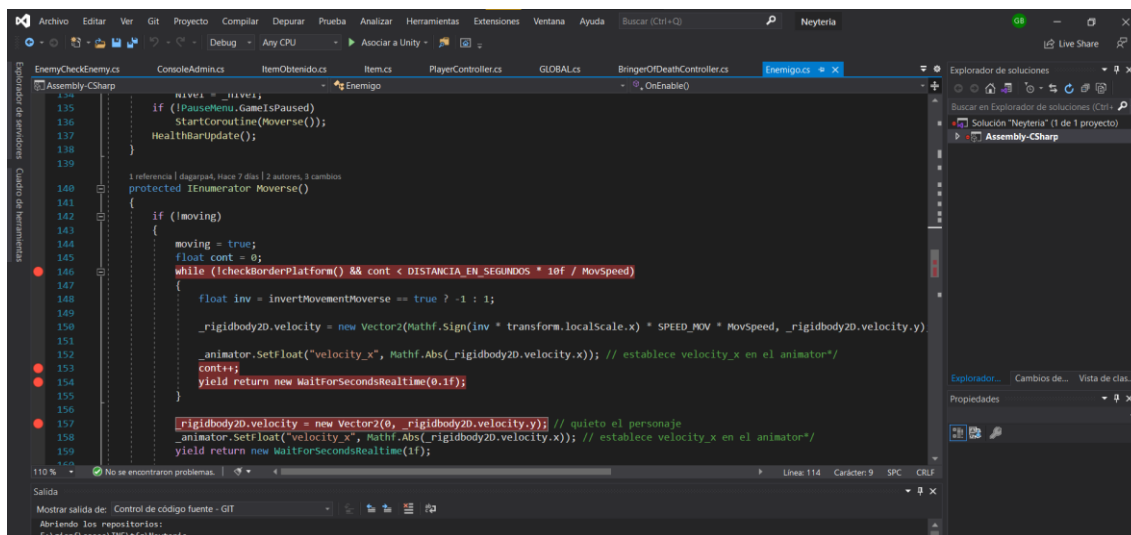


Figura 13. Scripts con *breakpoints* en lenguaje C# abiertos desde Visual Studio.

3.2.3. Git y GitHub

La herramienta Git, enlazada con GitHub, permite la coordinación del trabajo de varios integrantes de un equipo, de manera que cada integrante puede hacer cambios en el proyecto y subirlos a un repositorio donde se combinan todos los cambios que se ha hecho por parte del equipo. Además, también ofrece la posibilidad de crear distintas ramas donde subir los cambios sin que afecten a los del resto de ramas. Git permite catalogar el repositorio elegido como uno de Unity, que permite optimización a la hora de subir los cambios hechos en este. Esta es la herramienta que se ha utilizado en la asignatura para la colaboración entre alumnos al desarrollar un videojuego con Unity. En la Figura 14 se puede ver la aplicación “GitHub Desktop”, que hace posible la comodidad para subir, ver, y bajar cambios del repositorio. En la Figura 15 se puede ver el repositorio de GitHub desde el navegador, con todos los ficheros y carpetas del proyecto Unity.

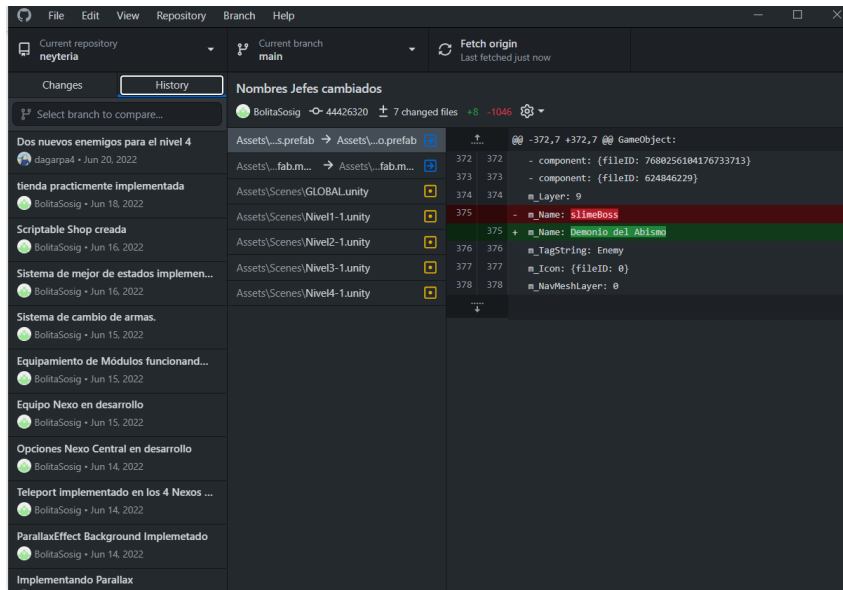


Figura 14. GitHub Desktop con lista de cambios.

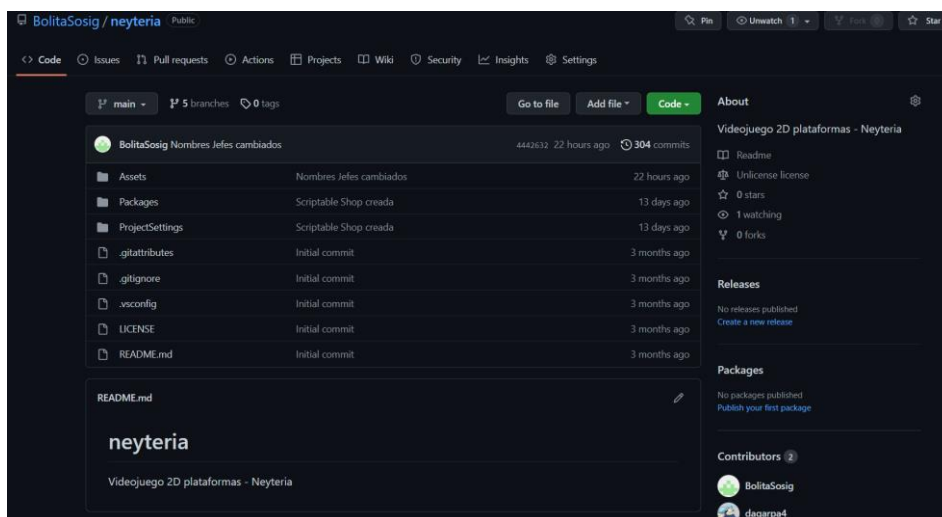


Figura 15. Proyecto subido a GitHub.

3.2.4. HackNPlan

HackNPlan es una herramienta web que permite controlar y planificar tareas a realizar para el desarrollo de un videojuego. Estas tareas deben pasar por cuatro fases: Planeada, En progreso, Probando y Completada. Estas se pueden clasificar en diferentes tipos como Programación, Arte, Diseño, Escritura, Marketing, Sonido, Ideas o Bug. También se le puede asignar un tiempo a cada tarea, para poder llevar el conteo de horas invertidas en el proyecto. HackNPlan es la herramienta perfecta, ya que está pensada para proyectos de desarrollo de videojuegos, pues tiene en cuenta

muchos aspectos que serían útiles para los desarrolladores, como vincular una cuenta de GitHub con el repositorio del proyecto. En la Figura 16 se muestra la Tabla “Menús e Interfaces” con sus tareas, algunas en progreso y otras ya completadas.

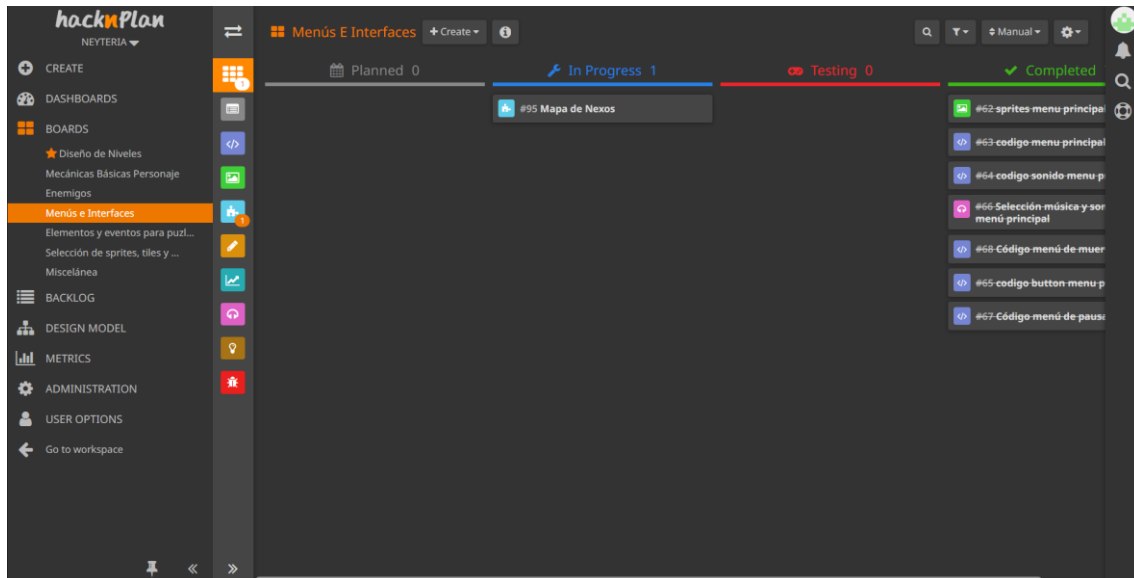


Figura 16. HackNPlan con algunas tareas.

4. DISEÑO DE LA SOLUCIÓN

4.1. Desarrollo del personaje

Cuando se quiere colocar cualquier objeto o script en una escena de Unity, se crea un *GameObject* en la jerarquía de objetos de esta. Esto también se aplica para el personaje que controlará el jugador, es decir, el personaje será un *GameObject* de la escena. Todo *GameObject* tiene un componente asociado, el *Transform*, que almacena la posición, rotación y escala del objeto en cuestión y algunas propiedades más, por lo que este componente será imprescindible para el control y desplazamiento del personaje. Por otro lado, los *GameObject* pueden tener otros objetos como hijos suyos en la jerarquía, haciendo que algunas propiedades de los hijos, como estar activo o las propiedades del *Transform*, dependan del objeto padre. Los *sprites* de todo el equipamiento del personaje, como las armas y los trajes, se encontrarán como hijos del *GameObject* principal del personaje. En la Figura 17 se puede ver la ordenación de los objetos que forman el personaje en la jerarquía de la escena de Unity. En la Figura 18 se puede ver al personaje principal en el juego.

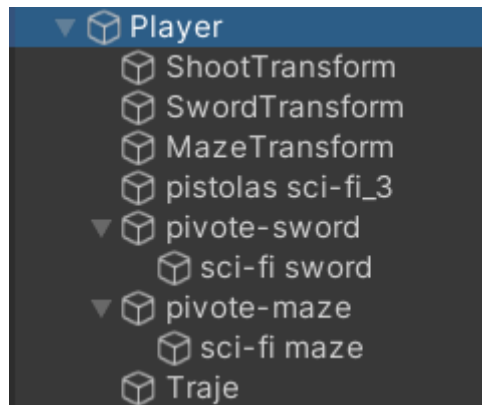


Figura 17. Jerarquía con los *GameObject* que forman el personaje.



Figura 18. Personaje principal en el juego.

El movimiento y control del personaje se realiza a través de un script que se vincula como componente del *GameObject*. De la misma manera, las colisiones, interacciones con el entorno, valor de los atributos del personaje, y más propiedades de este, se controlarán mediante scripts como componentes del objeto. Unity proporciona un inspector de objetos donde muestra todos los atributos, valores y propiedades de los scripts anclados a estos, haciendo posible la modificación de esos valores incluso cuando el juego se está ejecutando. No solo puede utilizarse para editar el valor de los atributos, sino también para visualizarlos y observar cómo van variando en tiempo de ejecución, como se ve en la Figura 19 y 20.

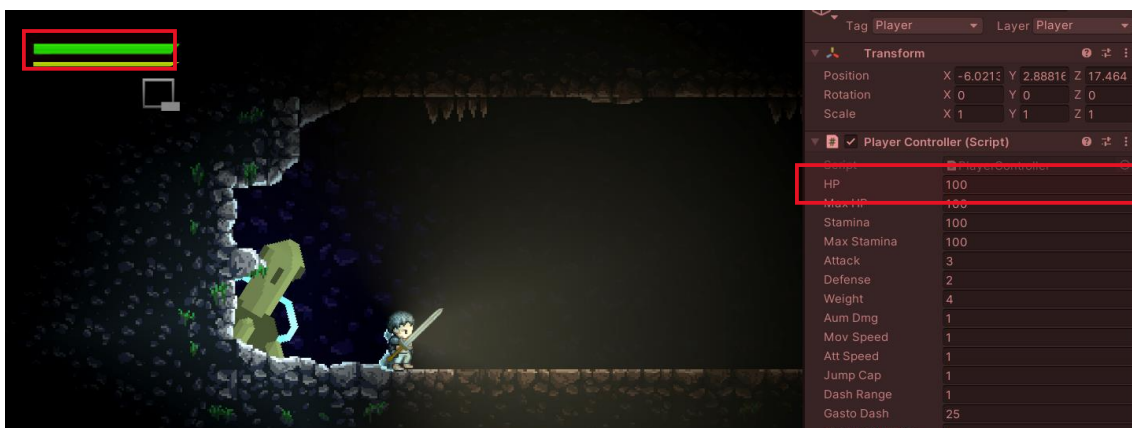


Figura 19. Inspector de Unity con atributos del personaje.



Figura 20. Inspector de Unity con atributos del personaje con la Salud aumentada.

Otra mecánica importante que posee el protagonista de “Neyteria” es la capacidad de ser invulnerable durante un pequeño intervalo de tiempo mientras está evadiendo. Esto es muy útil si, en momento justo en el que el personaje va a ser golpeado por algo, realizar una evasión y no recibir el daño. Dicha mecánica se ha programado con scripts que intervienen en los *frames* de la animación de la evasión, controlada por el *Animator* de Unity, calculando en que momentos de la animación debe ser invulnerable basándose en un atributo del personaje que define el porcentaje de intervalo de invulnerabilidad que tiene (*dash_range*).

4.2. Desarrollo del equipamiento

El protagonista de “Neyteria” precisa de un equipamiento para poder defenderse de las amenazas del mundo que le rodea. Para el desarrollo de las armas como se veía en la jerarquía de la Figura 17, se han colocado los *sprites* de estas en *GameObjects*, uno por cada tipo (Espada, Maza y Láser), de manera que solo este activo uno de los tres objetos, correspondiente al tipo del arma equipada. El script que controla todo el cambio entre armas se encuentra anclado al objeto padre del personaje.

Para el desarrollo de los trajes, la situación es prácticamente la misma que la de las armas, solo que únicamente con un *GameObject*, ya que no hay más que un tipo de traje. Las propiedades del

traje equipado y el control de la habilidad pasiva de este se dirigen desde un script específico que se encuentra anclado en el objeto padre del personaje.

En cuanto al desarrollo de los “Módulos del Ocaso”, estos no son visibles gráficamente con el personaje, así que no es necesario un *Sprite* hijo del objeto padre del personaje. Pero sí es necesario un script en este, que controle las habilidades activas de los módulos, su duración, su tiempo de enfriamiento y qué módulos tiene equipado en cada ranura, como se observa en la Figura 21. Los “Módulos del Ocaso” equipados con sus duraciones y tiempos de enfriamiento se pueden contemplar en el juego en la interfaz gráfica del usuario, debajo de la barra de resistencia, como se ve en la Figura 22.



Figura 21. Inspector de Unity con datos de los Módulos.



Figura 22. Información de los Módulos en el HUD.

4.3. Objetos

Los objetos se pueden conseguir tanto de los Suministradores como al derrotar enemigos, solo que estos últimos pueden ser específicos de este. Los objetos pueden ser: materiales, consumibles, o equipamiento (armas, trajes y módulos).

Los materiales son objetos que no son consumibles y mayormente son específicos de cada enemigo. Por ejemplo, un *Slime* puede dejarte al derrotarlo el objeto “Gotas de Slime”, y solo estos o enemigos derivados del *Slime* podrán dejar ese material. Los enemigos no solo ofrecen materiales específicos, sino otros recursos como monedas “Degiterio” y, todos los objetos que dejen al derrotarlos se generarán en la escena en forma de orbes con el icono del objeto que se

podrán recoger al pasar a través de ellos, como se observa en la Figura 23. Además, al obtener un objeto de cualquier manera, se mostrará un mensaje en la parte izquierda de la interfaz gráfica.

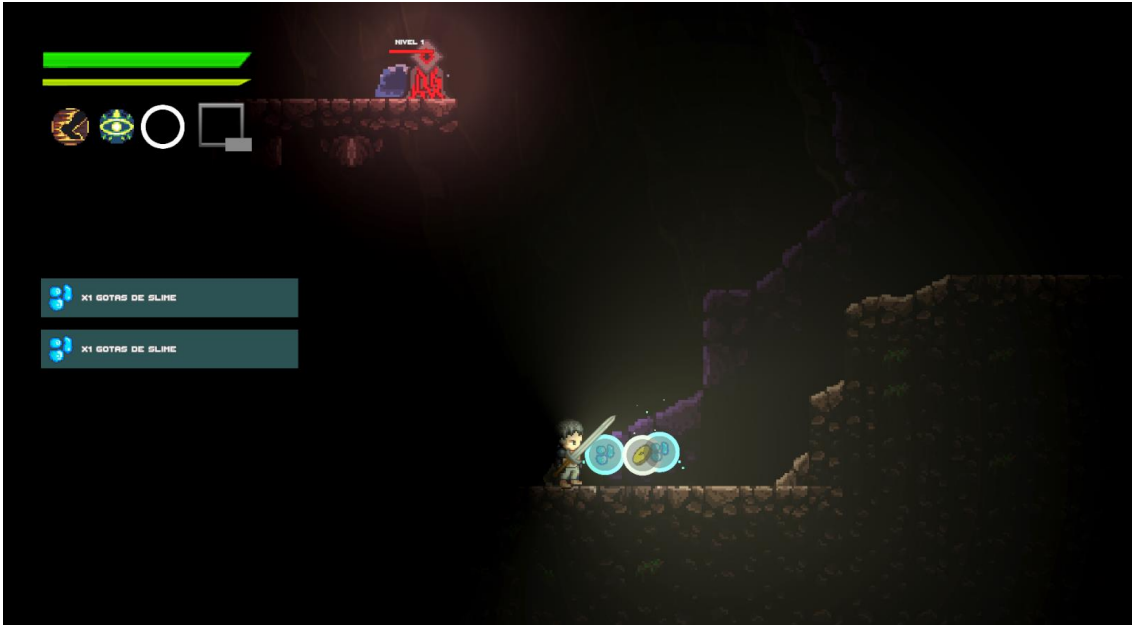


Figura 23. Orbes de objetos de *Slime* derrotados.

Toda la información de un objeto se encuentra definida en un script que hereda de *ScriptableObject* en lugar de *MonoBehaviour* como los anteriores. Los scripts del segundo tipo son aquellos que cuando los anclas a una *GameObject* se ejecutan en cada *frame* del juego, mientras que los primeros no, usándose como recipiente para almacenar datos. Una vez definido un *ScriptableObject*, se pueden crear ficheros que sean del tipo del script (en este caso de tipo “Item”) y almacenar así múltiples instancias del objeto, como se observa en la Figura 24.

Para objetos que sean “Item” pero con características especiales como armas, trajes y módulos, se crea un nuevo *ScriptableObject* para cada uno solo con los atributos y características adicionales y se hereda de “Item”, como se ve en la Figura 25. Esto es necesario para que los objetos materiales y consumibles no compartan características de equipamiento como el ataque o la defensa, además de para poder distinguirlos desde el código y poder programar distintos comportamientos para cada tipo de objeto. En la Figura 26 se puede observar el diagrama de clases UML con el objeto “Item” y todos los objetos que toman herencia de este.

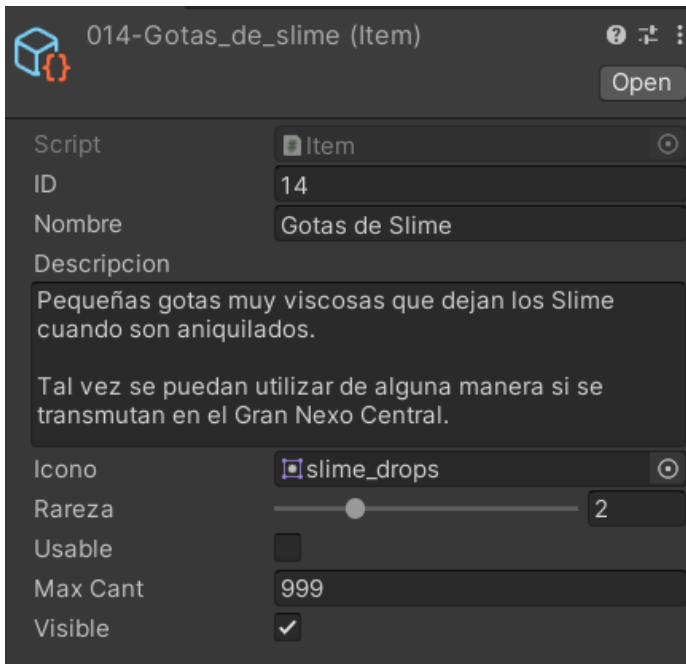


Figura 24. Atributos de un objeto “Item”.

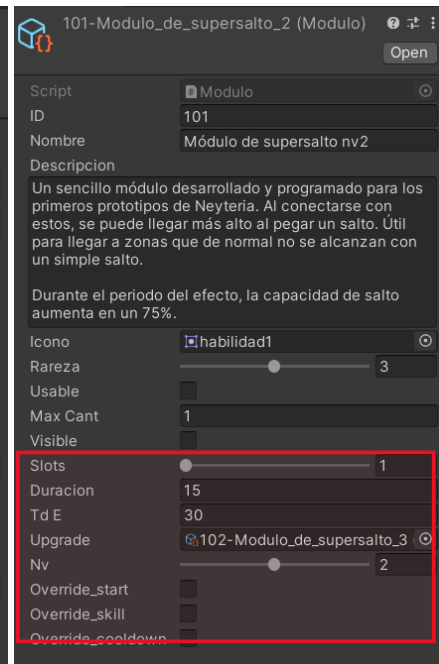


Figura 25. Atributos de un objeto “Modulo”.

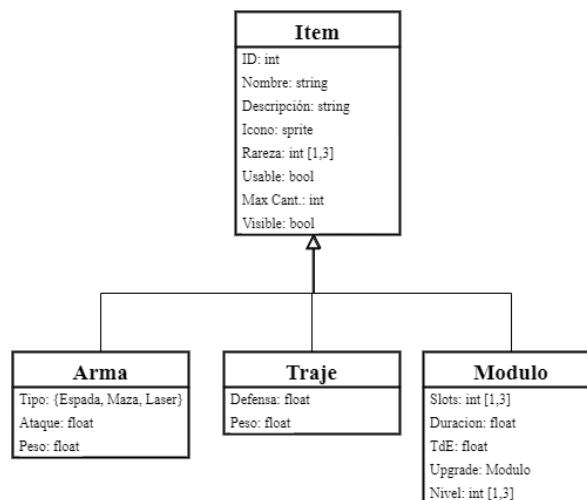


Figura 26. Diagrama de Clases UML de los objetos de tipo “Item”.

4.4. Animaciones

Las animaciones en un videojuego son muy importantes no solo para el movimiento del personaje o sus ataques, sino también para muchos otros efectos del entorno o de la interfaz. Todas las animaciones en Unity están controladas por un Animador y una máquina de estados que según qué condiciones sucedan en x momento, el animador controlará a la máquina de estados para que

elija una u otra animación a reproducir. Para el personaje principal, la máquina de estado de las animaciones controla las de *idle* o “quieto”, salto, evasión, caminar y el ataque con los tres tipos de arma.

En el animador siempre hay un único estado (naranja) que es desde el que se comienza a ejecutar la máquina de estados, es decir, todo el flujo que recorra partirá de ese estado. También existe un estado especial (azul) desde el que se accede en cualquier momento de la ejecución del animador, es decir, sea cual sea el estado que se esté ejecutando en ese momento, si las condiciones de alguna de las transiciones que parten del estado *Any State* se cumple, se ejecutará esa animación interrumpiendo la que estaba en curso. En la Figura 27, se puede observar la máquina de estados con las animaciones del personaje conectadas por transiciones con condiciones y la lista de parámetros del *Animator* utilizada para las condiciones.

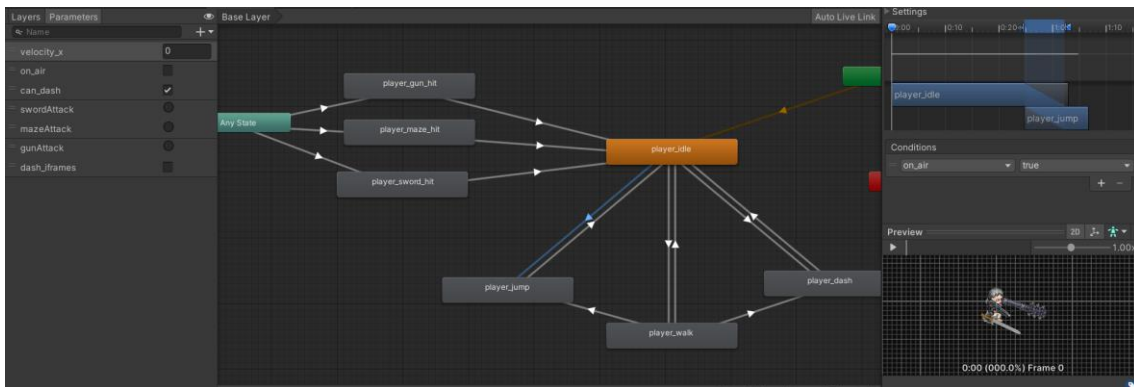


Figura 27. Máquina de estados del Personaje con sus animaciones.

Otro ejemplo distinto, como el de la Figura 28, sería el Nexo Central con su simple máquina de estados, ya que no se realizan animaciones de movimiento, sino solo de cambio de color. Concretamente, el Nexo Central se iluminará de color rojo cuando no se haya descubierto, azul cuando sí y verde cuando el jugador se esté teletransportando desde ese Nexo. Como se ve en la máquina de estados, solamente se puede llegar a la animación de teletransporte (*nexo_tp*) desde el estado del nexo activado (*nexo_on*) y no cuando esté por descubrir (*nexo_off*).

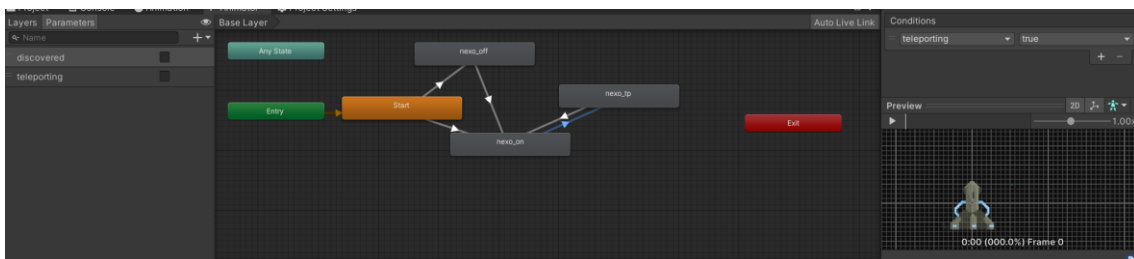


Figura 28. Máquina de estados del Nexo Central con sus animaciones.

4.5. Menús

Algo imprescindible para un videojuego es el desplazamiento del usuario entre los menús del juego. El primer menú que se muestra es el menú principal (Menú Principal), desde el que se puede entrar a jugar la partida (Start Game). Además, desde esa misma pantalla, se muestra la opción de cerrar y salir del juego (Exit Game) y se pueden acceder al menú de créditos (Credits) y al menú de ajustes (Settings), donde se puede ajustar el brillo y el volumen del juego.

Una vez dentro del juego, si se pulsa la tecla ESC se podrá acceder al menú de pausa (Pausa), desde el que se permite volver al menú principal y salir del juego directamente. Pulsando la tecla R se mostrará el menú de Equipamiento y Estados (Equipo / Estados), donde se podrá visualizar todo el equipamiento que el personaje posee, además de información sobre la pieza del equipo seleccionada que, al pulsar la tecla Z, se podrá alternar con información sobre las estadísticas del personaje en ese momento. Volviendo al juego, si se pulsa la tecla F, se abrirá el menú del inventario (Inventario), donde se contemplan todos los ítems consumibles y materiales que el personaje tiene guardados, además de mostrar información sobre el objeto seleccionado.

Desde el Nexa Central, una vez está activado, se mostrarán varias opciones que visualizarán un menú en la interfaz gráfica cada una. La primera opción es la de mostrar el menú de equipamiento, pero con la opción de cambiarse de equipo pulsando la tecla E al seleccionar un objeto (Equipo). La segunda opción es la de conectarse con otros nexos descubiertos y activados para poder transportarse hasta el seleccionado desde el mapa que se mostrará (Conectar). La tercera permite al usuario aumentar los niveles de los atributos del personaje a cambio de monedas, acompañado de información de las estadísticas del personaje (Estados). La última opción permite acceder a los suministros de todo tipo de objetos a cambio de otros (Suministros).

En la Figura 29, se puede ver un grafo dirigido con los menús del videojuego en cada nodo y con las aristas indicando las posibles transiciones entre menús.

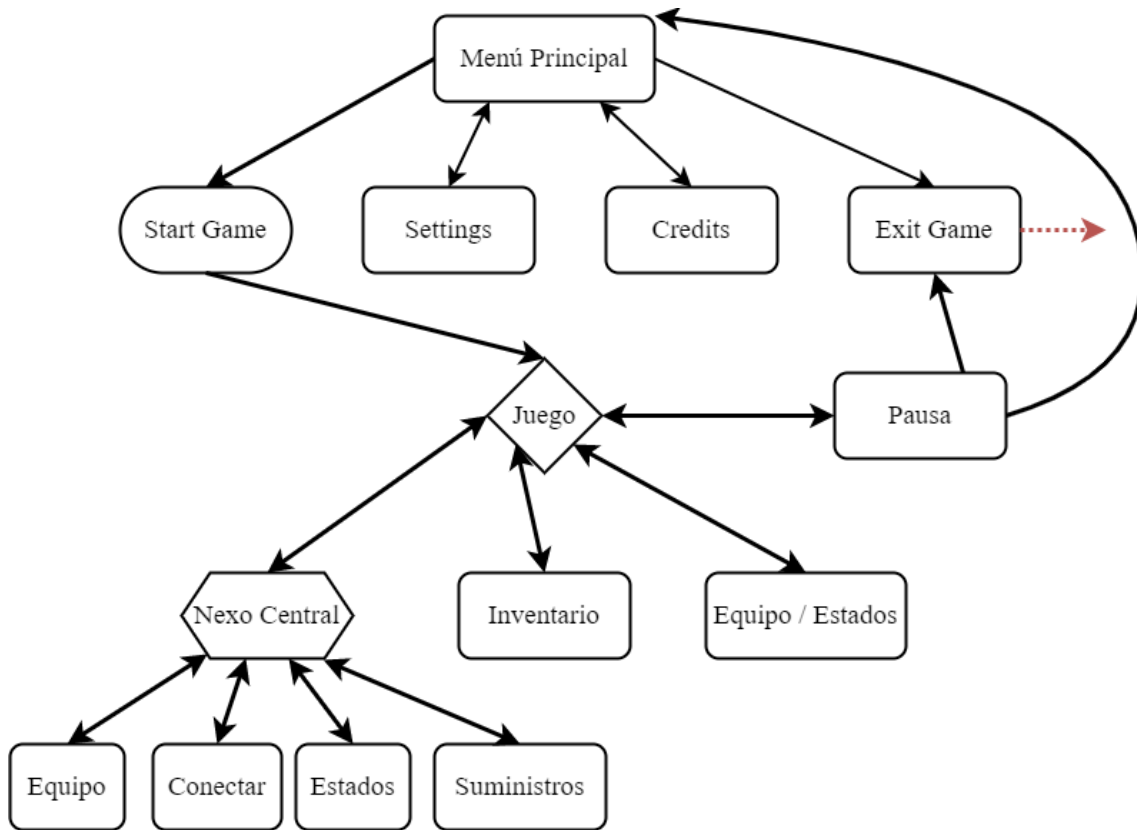


Figura 29. Grafo con todos los menús de “Neyteria”.

En la Figura 30 se puede ver un ejemplo de un menú del juego, concretamente el Menú Principal que se muestra cuando se inicia. En la Figura 31 se puede ver otro ejemplo, en este caso el menú del Inventario dentro de la partida.

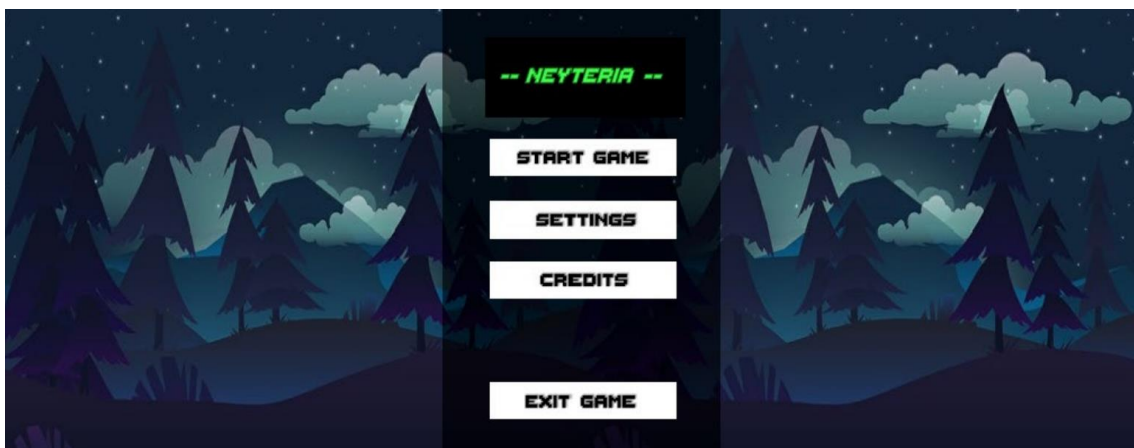


Figura 30. Menú principal de “Neyteria”.



Figura 31. Menú de Inventario del personaje principal con algunos objetos.

4.6. Enemigos

En “Neyteria” hay una gran diversidad de enemigos, algunos más feroces y otros más pasivos. Los enemigos tienen muchos atributos similares a los del personaje como son la salud, el ataque, la defensa, etc. Estos atributos son calculados en base al nivel del enemigo, el cual puede variar entre uno y ciento diez.

Para el desarrollo de los enemigos se ha creado un script general llamado Enemigo, donde se programa el comportamiento básico y pasivo de un enemigo, el daño que recibe cuando es dañado, al igual que el daño que provoca al personaje principal, y más funciones necesarias para que el enemigo sea funcional. Por otro lado, si se necesita hacer un enemigo con comportamientos distintos al básico, se le asigna un script personalizado para él de manera que este herede del script Enemigo, sobrescribiendo aquellas funciones que sean distintas. De esta manera, se pueden crear scripts específicos para cada enemigo sin tener que volver a programar en cada uno las funciones que sean iguales ni tener que volver a definir los mismos atributos para todos. En la Figura 32 se observa un esquema de los scripts de enemigos con sus correspondientes herencias.

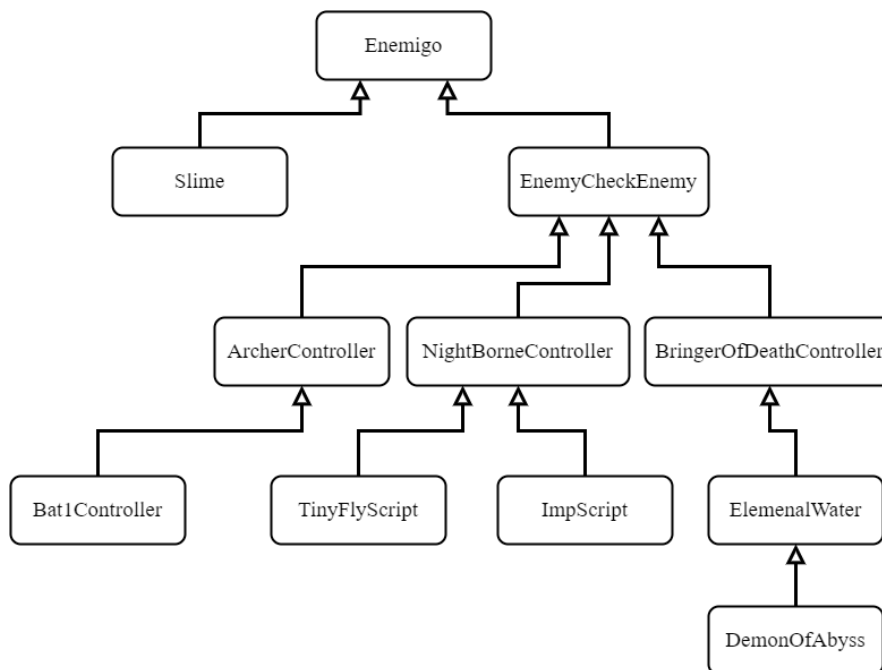


Figura 32. Diagrama de clases UML con la herencia de los scripts de enemigos.

La barra de salud de los enemigos se sitúa encima suyo junto al nivel de este mismo. En el caso de los jefes finales de cada nivel, su salud y nivel no se mostrarán encima del enemigo sino en la parte inferior del HUD, junto al nombre del jefe. Como se ve en la Figura 33, los enemigos normales tienen su barra de salud y nivel encima, en este caso dos “Slimes”, mientras que la información del jefe se muestra debajo, siendo este en la imagen el “Portador de la Muerte”.

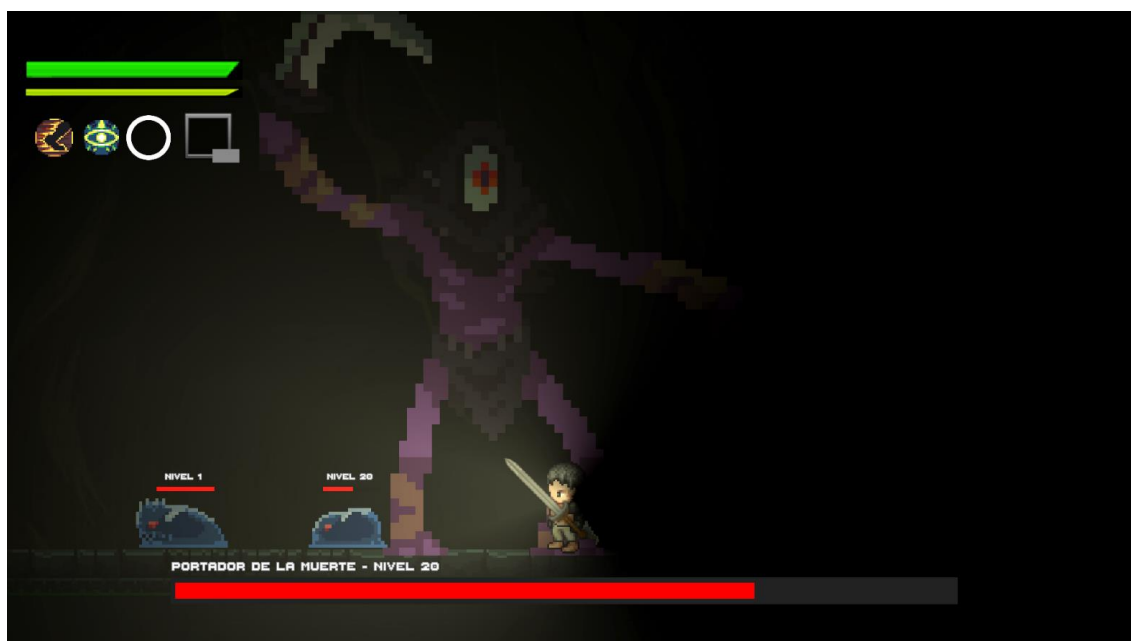


Figura 33. Nivel y barras de salud de los enemigos y jefes.

En la Figura 34 se observa el escalado de los atributos del mismo enemigo cuando el nivel cambia, en este caso de dos “Slimes”: uno con nivel uno y otro con nivel 20. Se puede ver que los atributos que escalan son la salud, el ataque, la defensa, la velocidad con la que atacan y la velocidad con la que se mueven.

Script	Slime	Script	Slime
Nivel	1	Nivel	20
HP	50	HP	62.86693
Max HP	50	Max HP	120.775
Stamina	100	Stamina	100
Attack	4	Attack	9.662
Defense	1	Defense	2.4155
Weight	1	Weight	1
Mov Speed	1	Mov Speed	1.69
Att Speed	1	Att Speed	1.69

Figura 34. Comparación entre los atributos de un “Slime” de nivel 1 frente a uno de nivel 20.

4.7. Niveles y optimización.

“Neyteria” cuenta con cuatro niveles conectados entre sí, donde para lograr pasar al siguiente nivel al que el jugador esté, debe derrotar al jefe de la zona. Las zonas son:

- **Grutas del Destino**, que se corresponde con el primer nivel. Estas grutas se encuentran sumergidas en la tierra del planeta “Oriquath” y la luz no llega hasta ellas. El jefe final de la zona para acceder al Nexo Central es el “Portador de la Muerte”.
- **Cuevas del Alba**, que se corresponde con el segundo nivel. Las cavernas de esta zona se acercan más a la superficie por lo que poseen algo de vegetación. El jefe final de la zona para acceder al Nexo Central es el “Torrente Elemental”.
- **Planos de Oriquath**, que se corresponde con la primera parte del tercer nivel. Formado por una cueva que sale a la superficie de “Oriquath”, donde se haya una gran explanada con el Árbol Ymrafihl. El jefe final de la zona para acceder al Nexo Central es el “Protector del Árbol Ymrafihl”.
- **Bosque Sagrado**, que se corresponde con la segunda parte del tercer nivel. Consta de varios caminos que atraviesan un bosque de árboles muertos que, hace mucho tiempo, florecían y deslumbraban de belleza. El jefe final de la zona para acceder al Nexo Central es el “Hechicero Oscuro”.

- **Profundidades del Ocaso**, que se corresponde con el cuarto nivel. Se encuentra en lo más profundo habitable del planeta, donde abunda la lava y los feroces enemigos. El jefe final de la zona es el “Guerrero del Abismo”, seguido por el “Demonio del Abismo”.

En la Figura 35 se muestran las cinco zonas de los cuatro niveles conectadas y cargadas al mismo tiempo. Se puede ver como los Nexos Centrales están unidos por las tuberías que permiten el desplazamiento del personaje entre niveles no contiguos. Cada zona se encuentra bordeada de un color indicando la separación entre niveles: Grutas del Destino de azul, Cuevas del Alba de amarillo, Planos de “Oriquath” de verde suave, Bosque Sagrado de verde fuerte y Profundidades del Ocaso de rojo.

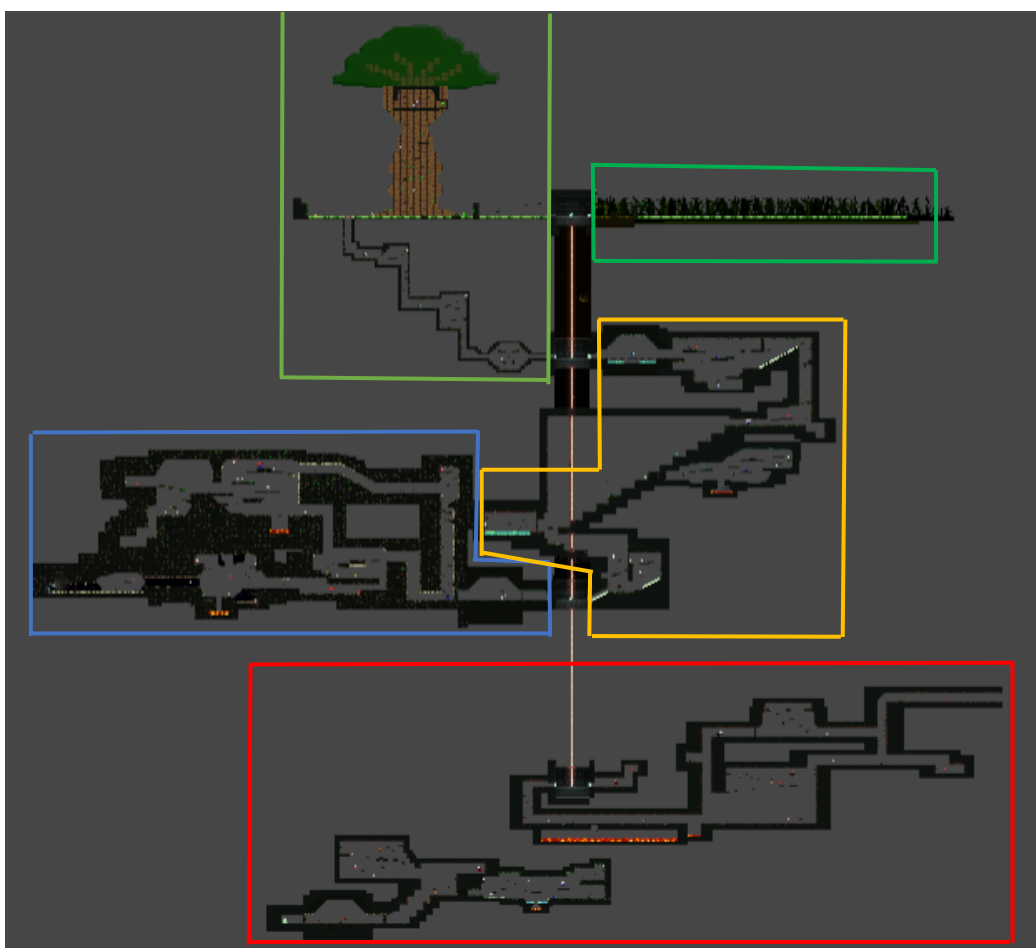


Figura 35. Niveles de “Neyteria” cargados en escena a la vez desde el editor de Unity.

Pero mantener todos los niveles cargados al mismo tiempo consumiría muchos recursos y en sistemas con bajo rendimiento podría no llegar a funcionar bien, ya que cada script de cada *GameObject* se está ejecutando continuamente en cada fotograma que reproduce el juego. Por

eso, la manera de optimizarlo para ganar rendimiento y que a ojos del jugador se note la continuidad que se ha usado e implementado en “Neyteria” consiste en mantener cargado en escena el nivel en el que se encuentra el jugador y, siempre y en todo momento, la escena de los Nexos centrales. Cuando el personaje que aproxime a la entrada a los Nexos de cada nivel, una colisión detectara que el personaje se está acercando al Nexo. En ese momento, se cargará también el nivel que continúa del otro lado del Nexo Central de manera que, cuando el personaje se encuentre en uno de estos, tanto el nivel anterior como el siguiente estén cargados.

4.8. Diálogos y mensajes.

La manera de comunicarle algo al jugador de manera escrita dentro de la partida del juego es a través de mensajes y diálogos. Algunos aparecerán automáticamente al pasar por ciertas zonas, como es el caso de los tutoriales del principio del nivel uno, donde los diálogos van saliendo a medida que se avanza indicando los controles del juego. Otros aparecerán sólo si son activados por el jugador manualmente, interaccionando con el entorno, como puede ser el caso de notas sueltas por el suelo o algún suministrador bloqueado.

En el HUD principal hay oculto un panel de diálogo vacío, ya que los textos de cada diálogo se encuentran en la jerarquía de *GameObjects* como objetos de juego con un script que contiene su funcionamiento y los distintos bloques de texto. Una vez se activa un diálogo, toda la información de los bloques de texto pasa al panel de diálogo del HUD y se muestra ante el jugador. En la Figura 36 se puede ver un ejemplo de un diálogo de tutorial. En la Figura 37 se observa un suministrador de objetos bloqueado por enemigos, siendo un diálogo lo que informa al jugador de eso una vez intenta desbloquearlo.



Figura 36. Diálogo del comienzo del juego que informa sobre cómo atacar.

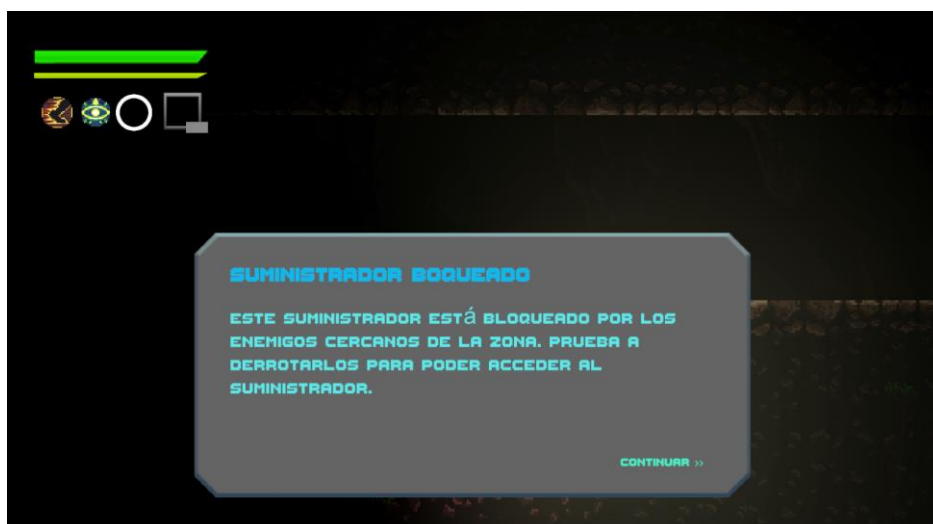


Figura 37. Diálogo de un Suministrador bloqueado.

También se pueden encontrar otro tipo de mensajes de texto que, en algunos casos, requieren de confirmación o negación por parte del jugador. Un ejemplo son los mensajes que aparecen cuando se intenta canjear algo en el Suministrador del Nexo Central, donde se preguntará si realmente se quiere obtener ese objeto a cambio de otro. Otro ejemplo sería cuando el jugador pretende subir de nivel alguno de los atributos del personaje, pues se le preguntará si esta seguro de subir ese atributo de nivel a cambio de monedas, teniendo la opción de rechazarlo y no realizar la operación, como se ve en la Figura 38.

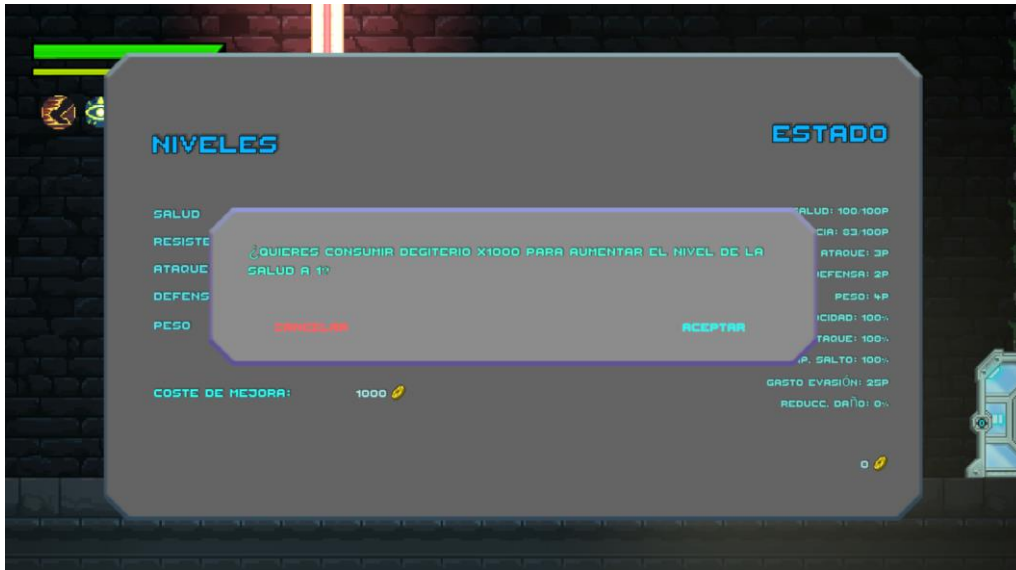


Figura 38. Mensaje de confirmación.

5. PRUEBAS Y TESTING

Para realizar un buen estudio de las pruebas del juego ofrecido al público, es necesario una cantidad de tiempo que no se ha presentado en este caso, pues por ello las pruebas y el testeo del juego se ha realizado a través de escenas de Unity preparadas para ello y los mismos desarrolladores han sido los encargados de hacerlas.

En estas pruebas se han hecho comprobaciones sobre las mecánicas del personaje principal y su correcta interacción con los enemigos, pruebas de menús funcionales y sin errores y otras pruebas de combate contra enemigos en solitario y en grupos mientras se modificaban sus atributos y los del personaje. Una funcionalidad perfecta para el testeo de un videojuego es la implementación de una línea de comandos que permita modificar o crear cualquier valor de cualquier entidad del juego sin necesidad de un editor, solamente con comandos, pudiendo así instanciar enemigos en tiempo real, recibir cualquier cantidad de cualquier objeto, modificar los atributos del personaje, etc.

En la Figura 39 se observa un ejemplo de escena de prueba donde se encuentran todos los enemigos del juego.

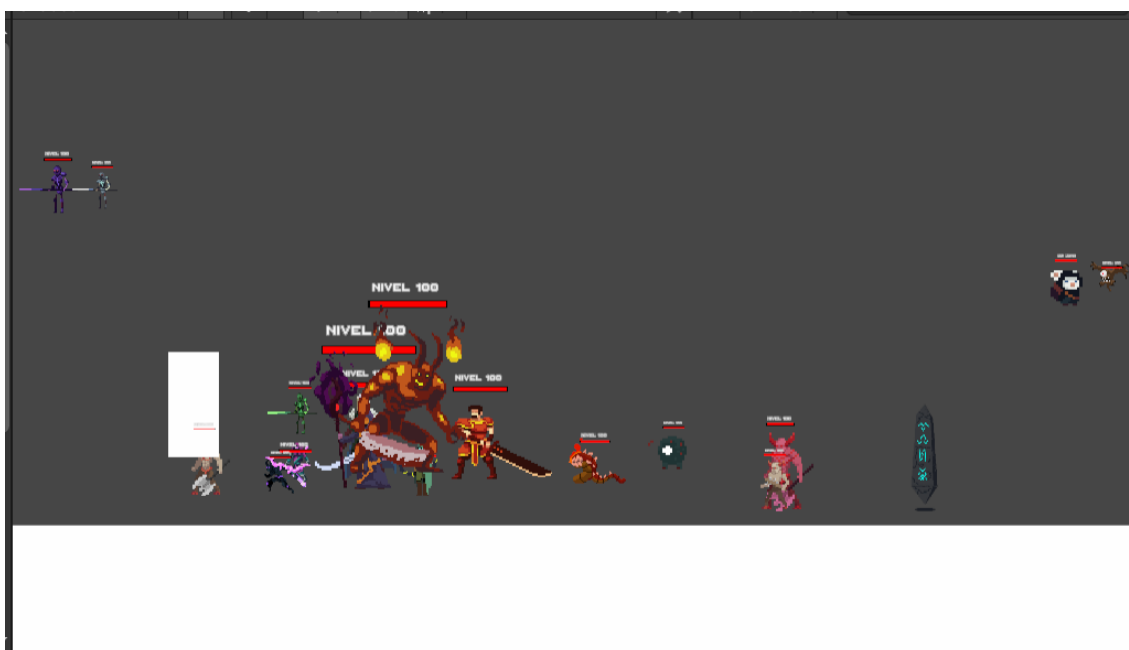


Figura 39. Escena de Unity para pruebas con todos los enemigos.

Por otro lado, una manera fácil de que el público pruebe el juego sin tener que subirlo a la web es presentárselo a conocidos cercanos, como la familia o a amigos y compañeros. Una vez probaron unas horas de juego, se les pasó una encuesta con varias preguntas para obtener *feedback* y contemplar qué fuertes y qué debilidades posee “Neyteria”.

Estas encuestas y preguntas se realizaron con 8 familiares y a amigos de los desarrolladores del juego, David Garrido y Gianfranco Pousa, obteniéndose así las siguientes estadísticas:

- Nivel de dificultad del videojuego (Figura 40):
 - Muy alto: 12,5%
 - Alto: 50%
 - Medio: 25%
 - Bajo: 12,5%
 - Nulo: 0%
 - NS/NC: 0%

Nivel de dificultad del videojuego

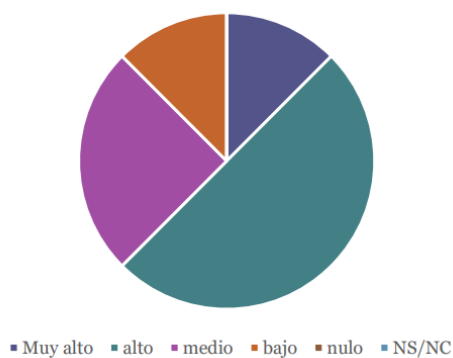


Figura 40. Gráfico circular con respuestas a “Nivel de dificultad del videojuego”.

- Movimiento del personaje (Figura 41):
 - Muy cómodo: 62,5%
 - Cómodo: 12,5%
 - Poco cómodo: 25%
 - Incómodo: 0%
 - Injugable: 0%

- NS/NC: 0%

Movimiento del personaje

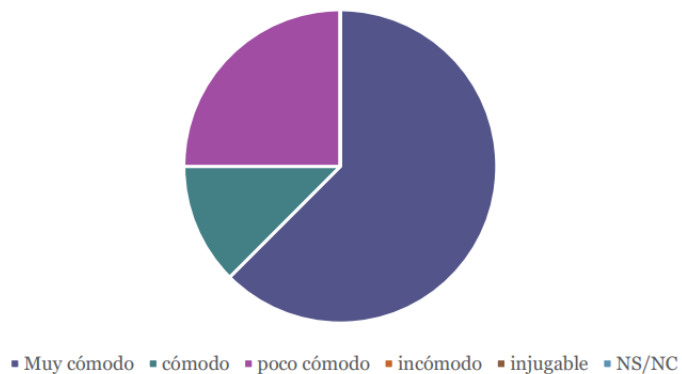


Figura 41. Gráfico circular con respuestas a “Movimiento del personaje”.

- Diseño del nivel (Figura 42):

- Genial: 0%
- Decente: 75%
- Pasable: 0%
- Mal: 12,5%
- Horrible: 12,5%
- NS/NC: 0%

Diseño de nivel

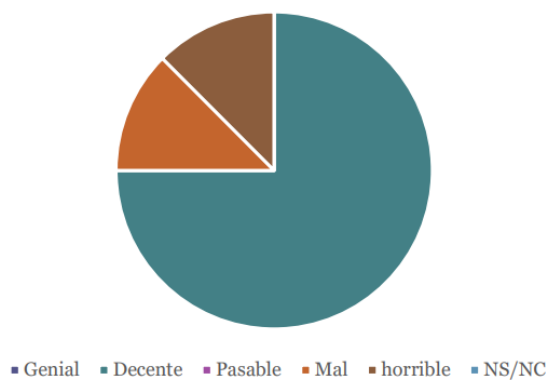


Figura 42. Gráfico circular con respuestas a “Diseño de nivel”.

- Entretenimiento del juego (Figura 43):

- Muy divertido: 25%
- Entretenido: 62,5%

- Ni sí ni no: 37,5%
- Aburrido: 12,5%
- Insoportable: 0%
- NS/NC: 0%

Entretenimiento del juego

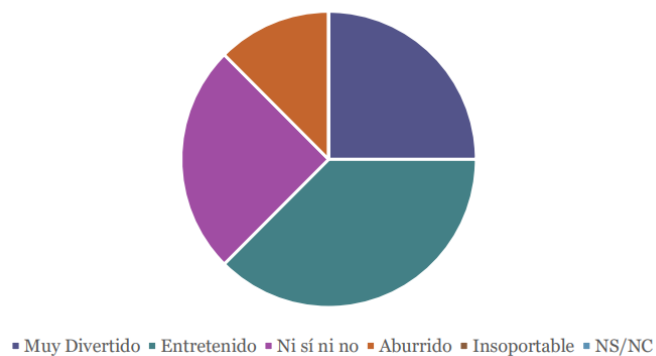


Figura 43. Gráfico circular con respuestas a “Entretenimiento del juego”.

Aun habiendo recibido *feedback* de personas ajenas al proyecto, se ha subido a la plataforma itch.io, donde se suben juegos “indie” que la gente puede descargarse el juego y opinar en la plataforma sobre este.

6. CONCLUSIONES

A pesar de trabajar a contrarreloj, con poco tiempo de desarrollo y con otras miles de tareas a realizar en la rutina, los objetivos propuestos en el planteamiento del proyecto han sido prácticamente alcanzados con éxito y con la idea en mente que David y yo teníamos. El único objetivo, si se puede considerar así, que no hemos logrado alcanzar, es el de poder trabajar con un diseñador gráfico que se encargue de las texturas y *sprites* del videojuego, ya que hemos tenido que recurrir a descargarlas de la web o apañarnos haciéndolas nosotros, que no es lo que teníamos pensado.

Pero aún así la funcionalidad del videojuego con todas sus características se ha conseguido lograr gracias a la colaboración que se ha tenido entre los desarrolladores y el apoyo que se nos ha proporcionado por parte de nuestro tutor y profesor de la asignatura “Desarrollo de Videojuegos 2D”, en la que también hemos aprendido aspectos necesarios e imprescindibles para poder crear un videojuego. La comunicación por “Teams” con nuestro tutor y la nuestra por plataformas de llamada como “Discord” ha sido eficaz y de gran ayuda para el avance constante del proyecto.

Otra dinámica que ha sido de gran ayuda para avanzar de manera cómoda y eficiente es la distribución estricta de trabajo entre los miembros del equipo de desarrolladores, ya que uno se encargaba de ciertos aspectos del juego en su totalidad y el otro de otros aspectos que, intentábamos que no influyesen entre ellos mucho, aunque en algunos casos era imposible, ya que el funcionamiento de un videojuego depende de cada una de las tareas que teníamos asignadas.

6.1. Relación del trabajo con los estudios cursados.

A lo largo del Grado en Ingeniería Informática se han cursado distintas asignaturas de las cuales el contenido enseñado es imprescindible para el desarrollo de un videojuego como con distintos lenguajes de programación para poder adaptarse fácilmente a otros.

Algunas de las asignaturas a destacar que han sido imprescindibles son:

- **Desarrollo de Videojuegos 2D**, donde se explican todas las funcionalidades de la herramienta Unity y la programación en lenguaje C#.
- **Gestión de proyectos**, donde se enseñan los métodos y pasos a seguir a la hora de realizar un proyecto en equipo, así como a planificar las tareas a realizar.
- **Ingeniería de Software**, en la que se proporcionó la experiencia para realizar correctamente un proyecto software con sus dependencias y cómo aplicar pruebas a este.
- **Interfaces Persona Computador**, donde se enseñó a diseñar correctamente una interfaz gráfica para el usuario en una aplicación.

Además de las asignaturas cursadas, se han aprovechado las siguientes competencias transversales:

- **Diseño y proyecto**, para llevar a cabo un proyecto correctamente.
- **Trabajo en equipo y liderazgo**, para poder entenderse y coordinarse entre los miembros del equipo de desarrolladores.
- **Análisis y resolución de problemas**, para poder analizar y desarrollar soluciones acertadas a los problemas que surgen durante el desarrollo del proyecto

7. TRABAJOS FUTUROS

A pesar de haber alcanzado los objetivos propuestos al comienzo del desarrollo del proyecto, es posible que alguno de ellos no se haya logrado con una mejor calidad o eficiencia. Es por eso por lo que “Neyteria” es un videojuego que se podría llegar a mejorar muchísimo rehaciendo algunas mecánicas o implementando nuevas e innovadoras y con más trampas y acertijos esparcidos por los niveles del mundo.

También se podrían hacer más niveles o incluso rehacer algunos de los que ya existen con nuevas texturas y *sprites* que embellezcan más al juego. Aunque si se pudiese, para el tema de los *sprites*, lo mejor sería tener en el equipo a un diseñador gráfico de B.B.A.A que tenga experiencia en realizar diseños digitales para videojuegos.

Además, se podría considerar añadir nuevos modos de juegos como un nivel que sea una sala con oleadas de monstruos cada vez más difíciles, para poder obtener equipamiento exclusivo, o algún sistema cooperativo donde otros usuarios puedan unirse a los mundos del resto.

8. REFERENCIAS

- [1] Frasca, G. (2009). Juego, videojuego y creación de sentido. Una introducción. Comunicación: revista Internacional de Comunicación Audiovisual, Publicidad y Estudios Culturales, 1 (7), 37-44.
<http://hdl.handle.net/11441/58039>
- [2] José Luis Eguia Gómez et al. (2013, marzo). Videojuegos: conceptos, historia y su potencial como herramientas para la educación. Revista de investigación. Editada por Área de Innovación y Desarrollo, S.L. <https://www.3ciencias.com/wp-content/uploads/2013/04/videojuegos.pdf>
- [3] Colaboradores de Wikipedia. (2022, 1 abril). Videojuego independiente. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Videojuego_independiente
- [4] Diana Díaz Montón. La traducción amateur de videojuegos al español (2011). TRANS. Revista de Traductología. (N.o 15). http://www.trans.uma.es/pdf/Trans_15/69-82.pdf
- [5] de Zayas, R. (2021, 5 febrero). 45 + 1 razones para traducir tu videojuego profesionalmente. Tatutrad. <https://tatutrad.net/5-1-razones-para-traducir-tu-videojuego-con-profesionales/>
- [6] Trigás Gallego, M. (2012). Metodología scrum. <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>
- [7] Mike Croucher. (2020). Academic Benefits of Using git and GitHub - Walking Randomly. <https://walkingrandomly.com/?p=6653>
- [8] Colaboradores de Wikipedia (2022) - Guía de los fundamentos para la dirección de proyectos PMBOK. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Gu%C3%ADa_de_los_fundamentos_para_la_direcci%C3%B3n_de_proyectos
- [9] Belli, S., & Raventós, C. L. (2008). Breve historia de los videojuegos. Athenea Digital. Revista de pensamiento e investigación social, (14), 159-179. <https://www.redalyc.org/pdf/537/53701409.pdf>
- [10] Plarium (2022). Juegos Metroidvania: todo lo que debes saber. <https://plarium.com/es/blog/metroidvania-games/>
- [11] AltexSoft (November, 2019): Non-functional Requirements: Examples, Types, How to Approach. <https://www.altexsoft.com/blog/non-functional-requirements/>
- [12] Adam (october, 2021). Functional Requirements in Software Engineering: Top Tips for Writing a Clear Document. <https://studiosoftware.com/blog/functional-requirements-insoftware-engineering-how-to-improve-project-performance>

- [13] Luis Matamala Belinchón (2021) ¿Qué son los Stakeholders y cómo se gestionan en los proyectos? IMF Smart Education. <https://blogs.imf-formacion.com/blog/mba/que-son-losstakeholders/>
- [14] Christian Leis (2022) Qué es y Cómo hacer Brainstorming. <https://aulacm.com/guia-hacerbrainstorming-generar-ideas-creativas/>
- [15] Milena González (2022) Análisis DAFO: Guía con Ejemplos <https://aulacm.com/analisisdafo-ejemplo-plantilla/>
- [16] Unity (2021) - Manual: Prefabs - Unity Technologies <https://docs.unity3d.com/es/530/Manual/Prefabs.html>
- [17] Unity (2021) - Manual: Prefabs Variants - Unity Technologies <https://docs.unity3d.com/Manual/PrefabVariants.html>
- [18] The best place for answers about Unity - Unity Answers <https://answers.unity.com/index.html>
- [19] Yúbal Fernández (2019). Qué es Github y qué es lo que le ofrece a los desarrolladores. Xataka. <https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores>
- [20] O G Glazunova et al 2021 J. Phys.: Conf. Ser. 1840 012030 - The effectiveness of GitHub cloud services for implementing a programming training project: students' point of view. <https://iopscience.iop.org/article/10.1088/1742-6596/1840/1/012030/pdf>
- [21] David Herron (2018) - What are the differences between Apache and GitHub? – Quora. <https://www.quora.com/What-are-the-differences-between-Apache-and-GitHub-Is-it-possiblethat-Apache-will-also-get-acquired-by-Microsoft-like-GitHub>

9. ANEXO

9.1. Licencias.

La gran mayoría de *sprites* que se han utilizado para el desarrollo de “Neyteria” han sido obtenidos de la web, concretamente de la plataforma itch.io. Los creadores de los que hemos utilizado son:

Créditos a **chierit** por *sprites* para Jefes: <https://itch.io/profile/chierit>

Créditos a **Clembod** por *sprites* para Jefes: <https://itch.io/profile/clembod>

Créditos a **LuizMelo** por *sprites* para Jefes y enemigos: <https://itch.io/profile/luizmelo>

Créditos a **kronovi** por *sprites* para enemigos: <https://itch.io/profile/darkpixel-kronovi>

Créditos a **CreativeKind** por *sprites* para enemigos, decoración y elementos del nivel: <https://itch.io/profile/creativekind>

Créditos a **Szadi art.** por *sprites* para enemigos y decoración: <https://itch.io/profile/szadiart>

Créditos a **AstroBob** por *sprites* para enemigos: <https://itch.io/profile/astrobob>

Créditos a **sanctumpixel** por *sprites* para enemigos: <https://itch.io/profile/sanctumpixel>

Créditos a **penusbmic** por *sprites* para enemigos: <https://itch.io/profile/penusbmic>

Créditos a **Anokolisa** por tiles para diseño de niveles: <https://itch.io/profile/anokolisa>

Créditos a **Pipoya** por *sprites* para elementos del nivel: <https://itch.io/profile/pipoya>

Créditos a **Tiny Worlds** por la fuente de texto: <https://itch.io/profile/tinyworlds>

Créditos a **adwitr** por *sprites* para el HUD: <https://itch.io/profile/adwitr>

Créditos a **wubs** por el generador de sonidos: <https://itch.io/profile/wubs>

9.2. Objetivos de Desarrollo Sostenible.

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
Fin de la pobreza				X
Hambre cero				X
Salud y bienestar			X	
Educación de calidad				X
Igualdad de género				X
Agua limpia y saneamiento				X
Energía asequible y no contaminante				X
Trabajo decente y crecimiento económico				X
Industria, innovación e infraestructuras				X
Reducción de las desigualdades				X
Ciudades y comunidades sostenibles				X
Producción y consumo responsables			X	
Acción por el clima				X
Vida submarina				X
Vida de ecosistemas terrestres				X
Paz, justicia e instituciones sólidas				X
Alianzas para lograr objetivos				X

En cuanto a los ODS de la tabla anterior marcados en Alto, Medio o Bajo:

- **Salud y bienestar:** los puzles y acertijos a resolver en el videojuego fortalece la inteligencia y la capacidad de pensar y resolver problemas, que a su vez ayuda a mantener una salud relativamente estable.
- **Producción y consumo responsables:** para la realización de este videojuego se ha intentado optimizar los recursos lo máximo posible para que el consumo sea mínimo en las plataformas para las que está pensado.

9.3. *Game Design Document.*

GAME DESIGN DOCUMENT



NEYTERIA

STRANGE WORLD

TABLA DE CONTENIDOS

ANÁLISIS DEL JUEGO	3
OBJETIVO DEL JUEGO	3
GÉNERO	3
PLATAFORMAS	3
PÚBLICO OBJETIVO	3
ANÁLISIS COMPETITIVO	4
REFERENTES	4
COMPETENCIA	4
CARACTERÍSTICAS DIFERENCIADORAS	4
HISTORIA Y PERSONAJES	5
JUGABILIDAD	23
VISIÓN GENERAL DEL JUEGO	23
EXPERIENCIA DEL JUGADOR	23
PAUTAS DE JUGABILIDAD	23
OBJETIVOS DEL JUEGO Y RECOMPENSAS	24
MECÁNICAS DEL JUEGO	30
EQUIPAMIENTO - SISTEMA DE PUNTUACIÓN DEL PERSONAJE	35
DISEÑO DE NIVEL	18
SONIDOS DEL JUEGO	45
ESQUEMA DE LOS CONTROLES	46
ESTÉTICA DEL JUEGO E INTERFAZ DEL USUARIO	46
TAREAS Y CALENDARIO	53

Análisis del Juego

Mata a tus enemigos y sobrevive para llegar al final en un entorno hostil usando tus capacidades de lucha y herramientas a tu alcance.

Deberás pasar a través de distintos niveles saltando, golpeando, corriendo, esquivando y usando tus habilidades y artefactos, para enfrentar y matar al boss de cada nivel.

Prepárate una buena equipación mediante el loot de los enemigos y obtén grandes recompensas en los puzzles ubicados entre cada nivel.

Objetivo del Juego

Neyteria, un juego de plataformas y acción para PC donde deberás usar tu capacidad, reflejos e inteligencia para poder superar exitosamente todos los niveles, de manera que avances en la historia y llegues al verdadero final.

Géneros

Plataformas, acción, aventura.

Plataformas

PC.

Público Objetivo

Jugadores apasionados de los retos difíciles de acción y plataformas, que estén acostumbrados a juegos de PC.



Análisis Competitivo

Referentes

Hollow knight, Celeste, Gris, Hyper Light Drifter. Juegos de plataformas donde haya enemigos, en general.

Competencia

Juegos como los nombrados como referentes y juegos de los géneros plataformas, acción y/o aventura.

Características Diferenciadoras

La idea principal es combinar los aspectos básicos de un plataformas en distintos niveles junto con características de acción y metiendo puzzles para ir cambiando el ritmo del juego.



Historia y personajes

Historia

CAPÍTULO I: El alba de un nuevo mundo.

En algún momento desconocido de la historia, el protagonista de esta se despierta, confuso y aturdido, en una habitación nada cuidada y prácticamente vacía, que está medio derrumbada con fragmentos de las paredes mezclados con la roca que rodea la habitación. Apenas hay una luz en el techo, parpadeando y a punto de colapsar. Cuando se vuelve más consciente, se da cuenta de que su brazo derecho está hecho de circuitos recubiertos de un material elástico pero rígido a la vez, que permite su manipulación a gusto sin problemas, y de tres cavidades circulares que parecen tener conexión con el interior del brazo. A pesar de no recordar absolutamente nada de su vida, le suena que nunca ha tenido un brazo robótico, sino uno normal como el de cualquier persona.

Entre la oscuridad puede distinguir restos robóticos y algunos huesos agrietados. Junto a estos, observa distintas armas para defenderse y considera que alguna de ellas le podría ser útil ante este entorno incierto. Desorientado y sin más opciones, decide avanzar y adentrarse en las cuevas subterráneas donde ha aparecido. Poco después se encuentra con criaturas abismales que tienen intención de atacarle. En ese momento, por instinto, saca el arma que había recogido al principio y se dispone a acabar con las criaturas.

Continúa avanzando por las Grutas del Destino hasta que se encuentra con una base muy cuidada, con un dispositivo muy grande con el que puede comunicarse a través del brazo robótico.

CAPÍTULO II: Memorias cálidas.

El protagonista se desenvuelve ágilmente en el campo de batalla mientras va haciendo suyo el control de los poderes del brazo biónico. Poco a poco, el protagonista y su brazo se van volviendo uno, y en el proceso va ganando capacidad de combate, habilidades y agilidad que le proporciona el mecanismo biónico.

Pero no menos importante, el protagonista va obteniendo también los recuerdos y memorias que yacen en el mecanismo futurista que dispone como brazo. Va teniendo visiones y obteniendo recuerdos, que aunque no son suyos, dan un inicio a la maduración de una personalidad y una identidad propia, como cualquier humano tendría.

De esta manera, se dispone a escapar de aquel lugar y vivir una nueva vida desde cero. Una vida tranquila, pero emocionante, donde se dedicará a buscar otros humanos, y de esta manera, poder conseguir tener una familia como en las memorias o en las visiones se describían. De esta manera sigue ascendiendo por la cueva hasta que se encuentra otra puerta deslizante, la cual abre con su brazo biónico y deja ver tras de sí una zona de la cueva, más vegetada y luminosa, indicando más vitalidad y una probabilidad más grande de que puedan haber humanos vivos en el mundo.

CAPÍTULO III: Dudas de la superficie.

El brazo biónico ya es parte del protagonista, la fusión es ahora total, y así lo son sus memorias y pensamientos. Ya no hay dificultad de comunicarse en combate, ni duda en la elección de habilidades. El protagonista ahora es, plenamente, un cyborg.

Esta unión se ha ido produciendo poco a poco, a lo largo de meses, por lo que el protagonista aún no se ha dado cuenta de este evento. Sin embargo, las visiones cada vez son más fuertes y ya no era capaz de distinguir qué es real y qué son recuerdos. Por fin había salido a la superficie y notaba los rayos de los soles en su rostro y su cuerpo. La ilusión que sentía por formar una familia debería estar ya en su punto máximo, pero no era así.

En su mente ya no yacía bondad e ilusión. Ahora había una nube dentro de su cabeza, que no le permitía sentir esas sensaciones. En cambio, solo sentía odio, rabia y sed de venganza. Sentimientos primitivos que una vez tuvo, o tal vez fueran del brazo biónico. Pero eso ya no importaba, el protagonista ya no era el mismo que una vez fue, un humano asustado y asolado en el interior de la cueva; ahora era un ser poderoso y ansioso por buscar más poder y cumplir su venganza a quienes lo atraparon.

CAPÍTULO IV: La verdad de Oriquath. Neyteria

La comodidad y tranquilidad que ofrecía la superficie ya no lo eran para el protagonista. No había nadie allí. Se sentía como un extraño en ese mundo, otra vez. Empezaba a notar que no debería estar ahí, o no debería haber existido nunca. Poco a poco iba cayendo más profundo en la locura. Ahora estaba vagando por el planeta, sin rumbo, solo propulsado por el poco odio que le quedaba, mientras iba, incontrolablemente, a quedarse vacío por dentro.

Pero un día, algo pasó. En sus viajes recorriendo la superficie, encontró un templo. Objetivamente, no tenía nada de especial, eran un montón de rocas y ladrillos formando estructuras, como había en todo el planeta. Pero al mirarlo de cerca, el protagonista empezó a tener sensaciones. Su corazón helado empezó a descongelarse, y unas lágrimas empezaron a brotar de sus ojos. “¿Por qué?”, se preguntaba extrañado tras esa reacción.

Dudoso, decidió entrar y echar un vistazo. Por dentro, veía cristaleras, imágenes, estatuas. Todo le daba nostalgia. Siguió mirando hasta que el brazo empezó a reaccionar. El brazo estaba brillando y le estaba

dirigiendo hacia el interior del templo. Las visiones no paraban, como si de mil vidas se trataran. Hasta que de repente pararon.


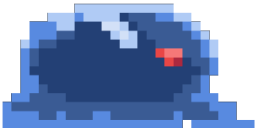
En el silencio y la oscuridad, apareció un ser de luz que iluminaba completamente el sitio. “Adelante, no tengas miedo”, le dijo el ser de luz. El protagonista dudoso, se acercó y se arrodilló ante él. “¿Quién eres, y qué es este lugar que me hace sentir así?”, preguntó firme mientras lo miraba. “Soy un espíritu y he venido a pedirte que salves el mundo” “Es el deber del elegido” respondió el ser de luz mientras le tocaba la cabeza.


Instantes después, el ser desapareció y el templo volvió a su oscuridad inicial. El protagonista, tras esta sagrada revelación se puso en marcha y bajó por un pasadizo del templo, en busca de su enemigo final.



CAPÍTULO V: El ocaso del viejo mundo.






Cada vez, el protagonista está más profundo en el planeta. La oscuridad es casi absoluta y seguramente nunca más vuelva a ver la luz del día. Cada vez hace más calor y el terreno es de un color más rojizo.

Personajes




Personaje	Descripción	Características	Información Miscelánea
Personaje principal 	Personaje jugable. Es el protagonista de la historia. El personaje principal es un humano medio cyborg, modificado con tecnología avanzada, con un brazo robótico que le permite, no solo tener una fuerza sobrehumana, sino también altos reflejos a la hora de combatir.	El protagonista podrá moverse por el mapa y usar habilidades y herramientas de supervivencia y combate cuando las necesite. Con astucia deberá atravesar todos los niveles y escapar del territorio hostil.	Ha acabado en un mundo desconocido y deberá desarrollar útiles habilidades de supervivencia para salvar su vida.
Slime (Primer enemigo) 	Primer enemigo que se encuentra el jugador. Es un slime que hace daño al entrar en contacto con el jugador.	Dispone con una inteligencia mínima para moverse de manera repetitiva hacia los lados, como si estuviera en guardia.	Se dice que viven en las cuevas y se alimentan del moho que yace entre las rocas.


<p>SlimeNvl2</p> 	<p>Slime del Nivel 2. Tiene más daño, vida y defensa que el original.</p>	<p>Dispone con una inteligencia mínima para moverse de manera repetitiva hacia los lados, como si estuviera en guardia.</p>	<p>Se dice que viven en las cuevas y se alimentan del moho que yace entre las rocas.</p>
<p>SlimeNvl4</p> 	<p>Slime del Nivel 2. Tiene más daño, vida y defensa que el original.</p>	<p>Dispone con una inteligencia mínima para moverse de manera repetitiva hacia los lados, como si estuviera en guardia.</p>	<p>Se dice que viven en las cuevas y se alimentan del moho que yace entre las rocas.</p>
<p>SlimeBerserker</p> 	<p>Slime en su versión Berserker. Gana tamaño, daño, vida y defensa.</p>	<p>Dispone con una inteligencia mínima para moverse de manera repetitiva hacia los lados, como si estuviera en guardia.</p>	<p>Se dice que viven en las cuevas y se alimentan del moho que yace entre las rocas.</p>
<p>Archer</p> 	<p>Uno de los primeros enemigos a distancia que se encuentra el jugador. Es un arquero arcano que dispara al jugador.</p>	<p>Su inteligencia le permite hacer guardia y vigilar. Cuando detecta al jugador lo persigue y le dispara. Si lo pierde de vista, vuelve a hacer guardia.</p>	<p>Los arqueros arcanos siguen vagando por el universo, saciando su sed de sangre.</p>
<p>Archer</p> 	<p>Archer del Nivel 2. Tiene más daño, vida y defensa que el original.</p>	<p>Su inteligencia le permite hacer guardia y vigilar. Cuando detecta al jugador lo persigue y le dispara. Si lo pierde de vista, vuelve a hacer guardia.</p>	<p>Los arqueros arcanos siguen vagando por el universo, saciando su sed de sangre.</p>



<p>ArcherHero</p> 	<p>Último enemigo a distancia que se encuentra el jugador. Es un arquero maldito que dispara al jugador.</p>	<p>Su inteligencia le permite hacer guardia y vigilar. Cuando detecta al jugador lo persigue y le dispara. Si lo pierde de vista, vuelve a hacer guardia.</p>	<p>Los arqueros malditos siguen cazando presas por la superficie o cuevas poco profundas por todo el planeta. Se dice que mantienen una dieta equilibrada.</p>
<p>FireWorm</p> 	<p>Enemigo con disparos a distancia. Se caracteriza por un gran daño en cada impacto.</p>	<p>Su inteligencia le permite hacer guardia y vigilar. Cuando detecta al jugador lo persigue y le dispara. Si lo pierde de vista, vuelve a hacer guardia.</p>	<p>Traído de los orígenes del planeta, es una de las formas de vida más antiguas y agresiva que se tenga registros.</p>
<p>FireWormLv14</p> 	<p>NightBorne del Nivel 4. Tiene más daño, vida y defensa que el original.</p>	<p>Su inteligencia le permite hacer guardia y vigilar. Cuando detecta al jugador lo persigue y le dispara. Si lo pierde de vista, vuelve a hacer guardia.</p>	<p>Traído de los orígenes del planeta, es una de las formas de vida más antiguas y agresiva que se tenga registros.</p>
<p>NightBorne</p> 	<p>Enemigo de combate cercano, con alta velocidad.</p>	<p>Su funcionamiento se basa en hacer guardia y vigilar. Cuando detecta al jugador lo persigue e intenta golpearle. Si lo pierde de vista, vuelve a hacer guardia.</p>	<p>Seres artificiales con un afán de consumir todo lo que esté a su alcance. Se alimentan de la energía vital de los seres vivos.</p>



<p>NightBorneNvl2</p> 	<p>NightBorne del Nivel 2. Tiene más daño, vida y defensa que el original.</p>	<p>Su funcionamiento se basa en hacer guardia y vigilar. Cuando detecta al jugador lo persigue e intenta golpearle. Si lo pierde de vista, vuelve a hacer guardia.</p>	<p>Seres artificiales con un afán de consumir todo lo que esté a su alcance. Se alimentan de la energía vital de los seres vivos.</p>
<p>NightBorneNvl3</p> 	<p>NightBorne del Nivel 3. Tiene más daño, vida y defensa que el original.</p>	<p>Su funcionamiento se basa en hacer guardia y vigilar. Cuando detecta al jugador lo persigue e intenta golpearle. Si lo pierde de vista, vuelve a hacer guardia.</p>	<p>Seres artificiales con un afán de consumir todo lo que esté a su alcance. Se alimentan de la energía vital de los seres vivos.</p>
<p>NightBorneNvl4</p> 	<p>NightBorne del Nivel 4. Tiene más daño, vida y defensa que el original.</p>	<p>Su funcionamiento se basa en hacer guardia y vigilar. Cuando detecta al jugador lo persigue e intenta golpearle. Si lo pierde de vista, vuelve a hacer guardia.</p>	<p>Seres artificiales con un afán de consumir todo lo que esté a su alcance. Se alimentan de la energía vital de los seres vivos.</p>
<p>NightBorneBerserker</p> 	<p>NightBorne en su versión Berserker. Gana tamaño, daño, vida y defensa.</p>	<p>Su funcionamiento se basa en hacer guardia y vigilar. Cuando detecta al jugador lo persigue e intenta golpearle. Si lo pierde de vista, vuelve a hacer guardia.</p>	<p>Seres artificiales con un afán de consumir todo lo que esté a su alcance. Se alimentan de la energía vital de los seres vivos.</p>
<p>HAO4</p> 	<p>Enemigo de combate cercano. No es muy rápido, pero suele ir en grupos grandes y tiene bastante rango.</p>	<p>Su funcionamiento se basa en hacer guardia y vigilar. Cuando detecta al jugador lo persigue e intenta golpearle. Si lo pierde de vista, vuelve a hacer guardia.</p>	<p>Almas perdidas que se niegan a descansar debido a su furia en sus vidas pasadas.</p>


<p>HAOLvl3</p> 	<p>HAO del Nivel 3. Tiene más daño, vida y defensa que el original.</p>	<p>Su funcionamiento se basa en hacer guardia y vigilar. Cuando detecta al jugador lo persigue e intenta golpearle. Si lo pierde de vista, vuelve a hacer guardia.</p>	<p>Almas perdidas que se niegan a descansar debido a su furia en sus vidas pasadas.</p>
<p>HAO4Berserker</p> 	<p>HAO4 en su versión Berserker. Gana tamaño, daño, vida y defensa.</p>	<p>Su funcionamiento se basa en hacer guardia y vigilar. Cuando detecta al jugador lo persigue e intenta golpearle. Si lo pierde de vista, vuelve a hacer guardia.</p>	<p>Almas perdidas que se niegan a descansar debido a su furia en sus vidas pasadas.</p>
<p>ImpMaze</p> 	<p>Enemigo de combate cercano. No dispone de mucha vida, pero suele ir en grupos grandes y tiene posibilidades de revivir.</p>	<p>Su funcionamiento se basa en hacer guardia y vigilar. Cuando detecta al jugador lo persigue e intenta golpearle. Si lo pierde de vista, vuelve a hacer guardia.</p> <p>Cuando su vida se reduce a cero, tiene una probabilidad de volver a la vida (con una penalización) y seguir luchando.</p>	<p>Demonio encontrado en las profundidades del planeta. Este recibe su energía del calor sagrado de Arold.</p>
<p>ImpAxe</p> 	<p>Enemigo de combate cercano. Dispone de mucha vida, y puede ir solo o en grupos grandes. Tiene posibilidades de revivir al ser abatido.</p>	<p>Su funcionamiento se basa en hacer guardia y vigilar. Cuando detecta al jugador lo persigue e intenta golpearle. Si lo pierde de vista, vuelve a hacer guardia.</p> <p>Cuando su vida se reduce a cero, tiene una probabilidad de volver a la vida (con una penalización) y seguir luchando.</p>	<p>Demonio encontrado en las profundidades del planeta. Este recibe su energía del calor sagrado de Arold. Suelen ser los líderes del grupo.</p>

<p>ImpAxeBerserker</p> 	<p>ImpAxe en su versión Berserker. Gana tamaño, daño, vida y defensa.</p>	<p>Su funcionamiento se basa en hacer guardia y vigilar. Cuando detecta al jugador lo persigue e intenta golpearle. Si lo pierde de vista, vuelve a hacer guardia.</p> <p>Cuando su vida se reduce a cero, tiene una probabilidad de volver a la vida (con una penalización) y seguir luchando.</p>	<p>Demonio encontrado en las profundidades del planeta. Este recibe su energía del calor sagrado de Arold. Suelen ser los líderes del grupo.</p>
<p>Bat1</p> 	<p>Enemigo volador de combate lejano. No es muy rápido, pero vuela y respeta las distancias.</p>	<p>Su funcionamiento se basa en hacer guardia y vigilar. Cuando detecta al jugador lo persigue e intenta golpearle. Si lo pierde de vista, vuelve a hacer guardia.</p> <p>Tiene un sistema para no acercarse demasiado al usuario, por lo que no se quedará ni muy lejos ni muy cerca.</p>	<p>Seres corrompidos que habitan suelen habitar las profundas cuevas profundas. Son bastante territoriales e inteligentes.</p>
<p>Bat2</p> 	<p>Enemigo volador de combate lejano. No es muy rápido, pero vuela y respeta las distancias. Esta versión dispone de mayor tamaño y proyectiles con seguimiento.</p>	<p>Su funcionamiento se basa en hacer guardia y vigilar. Cuando detecta al jugador lo persigue e intenta golpearle. Si lo pierde de vista, vuelve a hacer guardia.</p> <p>Tiene un sistema para no acercarse demasiado al usuario, por lo que no se quedará ni muy lejos ni muy cerca.</p>	<p>Seres corrompidos que habitan suelen habitar las profundas cuevas profundas. Son bastante territoriales e inteligentes. Suele ser el jefe de la manada.</p>




<p>TinyFly</p> 	<p>Enemigo volador de combate cercano. Posee una alta velocidad, volando reduce las distancias con el jugador.</p>	<p>Su funcionamiento se basa en hacer guardia y vigilar. Cuando detecta al jugador lo persigue e intenta golpearle. Si lo pierde de vista, vuelve a hacer guardia.</p>	<p>Este animal es uno de los más longevos del planeta y se alimenta de otros organismos a su alcance.</p>
<p>TinyFlyLv4</p> 	<p>TinyFly del Nivel 4. Tiene más velocidad, daño, vida y defensa que el original</p>	<p>Su funcionamiento se basa en hacer guardia y vigilar. Cuando detecta al jugador lo persigue e intenta golpearle. Si lo pierde de vista, vuelve a hacer guardia.</p>	<p>Este animal es uno de los más longevos del planeta y se alimenta de otros organismos a su alcance.</p>
<p>Bringer Of Death</p> 	<p>Enemigo Jefe del nivel 1. Combate cercano y habilidades a distancia. Invoca a otros enemigos del nivel 1.</p>	<p>El funcionamiento inicial es de hacer guardia esperando al jugador. Dispone de un poderoso ataque cercano y de rápidos ataques a distancia que impactan por encima del jugador. Dispone de 3 fases. La primera se trata del funcionamiento anterior. En la segunda (su vida baja de la mitad) invoca a 4 enemigos para ayudarlo en su pelea. En la tercera fase (su vida baja del 20%) invoca a 6 enemigos en el combate.</p>	<p>Ser maldito que vive en las profundidades del planeta. Con su legendaria arma, la segadora de Sagoh, y su vestimenta de Dal'avar, concede la muerte a todos ellos que busquen su perdición.</p>
<p>Elemental Water</p>	<p>Enemigo Jefe del nivel 2. Combate cercano y habilidades a distancia. Invoca a otros enemigos del nivel 2.</p>	<p>El funcionamiento inicial es de hacer guardia esperando al jugador. Dispone de un poderoso ataque cercano y de rápidos ataques a distancia que impactan por encima del jugador.</p>	<p>Diosa lorelei que vive escondida en las profundidades de la cueva. Porta una legendaria arma, el cetro de Talorian, posee el don para controlar</p>





		<p>Además dispone de 2 habilidades que se desbloquean al partir de la fase dos. La primera de ellas es un golpe a media distancia; y la segunda de ellas es un golpe cercano. Ambas son en área. Dispone de 3 fases. La primera dispone de un funcionamiento básico, con una forma determinada de andar, el ataque cercano y los ataques a distancia. En la segunda (su vida baja de la mitad) invoca a 4 enemigos para ayudarlo en su pelea, y desbloquea las dos nuevas habilidades. También se moverá de una forma más agitada. En la tercera fase (su vida baja del 20%) invoca a 6 enemigos en el combate.</p>	<p>las aguas y se cubre con su vestimenta de Princesa Lorelei. Con su enorme piedad, concede una paz tranquila a todos aquellos que no son capaces de cesar su agonía y quedan encantados con su melodía.</p>
<p>ElectricDefendant</p> 	<p>Enemigo Jefe de mitad del nivel 3. Combate cercano y habilidades a distancia. Invoca a otros enemigos del nivel 3.</p>	<p>Mismo funcionamiento que ElementalWater, pero con habilidades distintas.</p>	<p>Con sus milenios de antigüedad, este ser mantiene seguro y protegido el árbol madre de yygnail. Mediante el uso de su manto legendario, el protector de Afhén, y su sagrada espada, el brazo eléctrico de Railr, mantendrá alejado a todo aquel que busque dañar el bosque sagrado.</p>

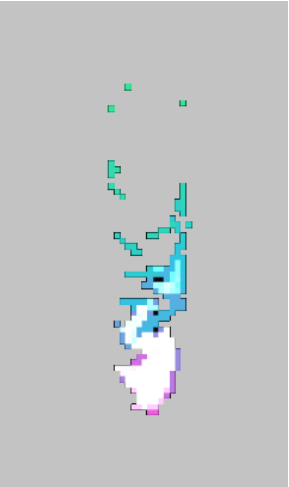


<p>EvilWizard</p> 	<p>Enemigo Jefe final del nivel 3. Combate cercano y habilidades a distancia. Invoca a otros enemigos del nivel 3.</p>	<p>Mismo funcionamiento que ElementalWater, pero con habilidades distintas.</p>	<p>La traición y soledad que sufrió a lo largo de su vida, convirtió a este pobre clérigo en un ser oscuro y malévolo que se desvió de la fe de Afhén. Mediante el uso del hábito de Rohim y el cetro de mal'ar, este ser se encargará de llevar desdicha y desasosiego a todo aquel que ose entrar en las profundidades del bosque de la noche.</p>
<p>DemonOfAbys</p> 	<p>Enemigo Jefe final del nivel 4. Combate cercano y habilidades a distancia. Invoca a otros enemigos del nivel 4.</p>	<p>Además del funcionamiento desarrollado en ElementalWater, este enemigo cuenta con una habilidad de evasión de daño. Cuando su vida llega a cero, invoca al enemigo SlimeBoss en la posición de muerte.</p>	<p>Súbdito de la orden del maligno, también apodado como Malzahar, que defiende la zona más profunda del reino del mal. Porta la espada maldita de sasog, y la armadura demoníaca de kat'lon, para llevar a un infierno terrenal a todo aquel que se adentre en este terreno maldito.</p>

<p>SlimeBoss</p> 	<p>Enemigo Jefe final Principal del nivel 4. Combate cercano y habilidades a distancia. Invoca a otros enemigos de alto calibre.</p>	<p>Mismo funcionamiento que ElementalWater, pero con habilidades distintas.</p> <p>Cuenta con habilidades más potentes e invocaciones de enemigos berserker.</p>	<p>Monstruo espiritual maldito, que se imbuje en los cuerpos de los hombres mediante la maldición de Malzaha. Este espíritu legendario y mitológico lleva a su final a todo ser y terreno que se interponga en su camino cuando es invocado en el mundo terrenal.</p>
--	--	--	---

Proyectiles

Proyectil	Descripción	Características	Información Miscelánea
<p>Bullet</p> 	<p>Proyectil que es lanzado por el personaje principal Player mediante el cañón láser.</p>	<p>El proyectil colisionará con enemigos y elementos del entorno.</p>	<p>Una de las balas básicas de energía de neutrones. Simple, pero potente.</p>
<p>Arrow</p> 	<p>Proyectil que es lanzado por el enemigo Archer.</p>	<p>El proyectil colisionará con el Player y elementos del entorno.</p>	<p>Flechas de metal inyectadas con energía de neutrones. Rápida y dolorosa potente.</p>
<p>Arrow</p> 	<p>Proyectil que es lanzado por el enemigo ArcherHero.</p>	<p>El proyectil colisionará con el Player y elementos del entorno.</p>	<p>Flechas de metal inyectadas con energía de neutrones. Rápida y dolorosa potente.</p>

<p>FireWormProjectile</p> 	<p>Proyectil que es lanzado por el enemigo FireWorm.</p>	<p>El proyectil colisionará con el Player y elementos del entorno.</p>	<p>Trozos de magma y ácido, convertidos en arma arrojadiza.</p>
<p>BatProjectile</p> 	<p>Proyectil que es lanzado por los enemigos Bat1 y Bat2.</p>	<p>El proyectil colisionará con el Player y elementos del entorno. Puede tener seguimiento o no del Player.</p>	<p>Trozo de carne y hueso, desprendido del cuerpo del huésped y los desechos de sus presas.</p>
<p>BringerOfDeathProjectile</p> 	<p>Proyectil que es invocado por el enemigo BringerOfDeath.</p>	<p>El proyectil solo colisionará con el Player. Este proyectil contiene animación. Solo en el momento idóneo (clímax) se producirá el daño al Player.</p>	<p>Energía oscura, obtenida de las almas perdidas que perecen en las profundidades.</p>
<p>ElementalWaterProjectile</p> 	<p>Proyectil que es invocado por el enemigo ElementalWater.</p>	<p>El proyectil solo colisionará con el Player. Este proyectil contiene animación. Solo en el momento idóneo (clímax) se producirá el daño al Player.</p>	<p>Porciones de almas y energía de agua y luna que forman una especie de fuego, que arde y quema todo aquello que causa dolor a los hombres.</p>

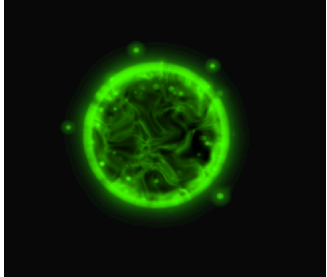
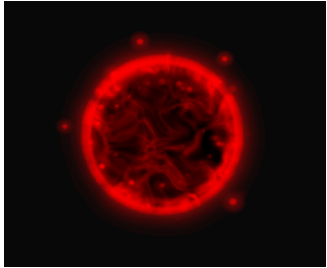



<p>ElectricDefendantProjectile</p> 	<p>Proyector que es invocado por el enemigo ElectricDefendant.</p>	<p>El proyectil solo colisionará con el Player. Este proyectil contiene animación. Solo en el momento idóneo (clímax) se producirá el daño al Player.</p>	<p>Fuerza eléctrica de los cielos canalizada en un punto cálido, donde se purifican las dudas y los males.</p>
<p>EvilWizardProjectile</p> 	<p>Proyector que es invocado por el enemigo EvilWizard.</p>	<p>El proyectil solo colisionará con el Player. Este proyectil contiene animación. Solo en el momento idóneo (clímax) se producirá el daño al Player.</p>	<p>Energía maldita y sagrada que interactúan en un punto del espacio, luchando para acabar la una con la otra.</p>
<p>DemonOfAbyssProjectile</p> 	<p>Proyector que es invocado por los enemigos DemonOfAbyss y SlimeBoss.</p>	<p>El proyectil solo colisionará con el Player. Este proyectil contiene animación. Solo en el momento idóneo (clímax) se producirá el daño al Player.</p>	<p>Brasas candentes e ígneas invocadas en un punto, que abrasan e incineran cualquier trozo tipo de materia que encuentre a su alcance.</p>

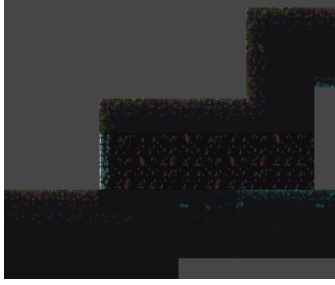

Decoración e Interactuables

Entidad	Descripción	Características	Información Miscelánea
---------	-------------	-----------------	------------------------

<p>Puerta SciFi</p> 	<p>Objeto decorativo</p>	<p>Reacciona al movimiento del jugador cuando es atravesado.</p>	<p>Puertas de alta tecnología que una vez fueron usadas por una civilización adelantada.</p>
<p>Obelisco</p> 	<p>Objeto decorativo</p>	<p>Ilumina el mapa y ofrece una visión temática del nivel al jugador.</p>	<p>Elemento sagrado y de culto, que fue construido como símbolo de veneración de una civilización anterior.</p>
<p>ToggleButton</p> 	<p>Objeto interactuable. El jugador puede usarlo mediante el uso de la tecla Barra Espaciadora.</p>	<p>Permite la generación u ocultación de terreno y plataformas del nivel.</p>	<p>Mecanismos de la era antigua que fueron usados para la mejora de vida de la civilización.</p>
<p>ToggleButtonBlue</p> 	<p>Objeto interactuable. El jugador puede usarlo mediante el uso de la tecla Barra Espaciadora.</p>	<p>Permite la activación y desactivación de elementos de puzles del nivel.</p>	<p>Mecanismos de la era antigua que fueron usados para la mejora de vida de la civilización.</p>

<p>Suministrador</p> 	<p>Objeto interactuable. El jugador puede usarlo mediante el uso de la tecla Barra Espaciadora.</p>	<p>Ofrece al jugador una serie y cantidad de materiales útiles para su uso en el sistema de inventario y tienda del juego.</p>	<p>Estructura antigua que mezcla conocimientos de tecnologías modernas y ancestrales.</p>
<p>Agua</p> 	<p>Objeto interactuable. El jugador reaccionará al elemento mediante el contacto con él.</p>	<p>El jugador será desplazado en la dirección en la que se mueva dicha corriente. Esta permitirá o no el movimiento del personaje mientras esté dentro de ella.</p>	<p>Corriente de agua encontrada en ciertas alturas de las cuevas y el planeta.</p>
<p>Lava</p> 	<p>Objeto interactuable. El jugador reaccionará al elemento mediante el contacto con él.</p>	<p>El jugador recibirá gran daño de forma frecuente mientras siga en contacto con este elemento.</p>	<p>Río de lava procedente de las profundidades del abismo y del centro del planeta.</p>
<p>Ventilador Ascendente</p> 	<p>Objeto interactuable. El jugador reaccionará al elemento mediante el contacto con él.</p>	<p>El jugador será desplazado en sentido ascendente mientras siga dentro de la corriente de aire.</p>	<p>Objeto mecánico diseñado por algún tipo de civilización no muy antigua.</p>
<p>Ventilador Ascendente Activable</p> 	<p>Objeto interactuable. El jugador reaccionará al elemento mediante el contacto con él.</p>	<p>El jugador será desplazado en sentido ascendente mientras siga dentro de la corriente de aire. Este ventilador estará desactivado, y podrá ser activado o desactivado mediante otros elementos del nivel.</p>	<p>Objeto mecánico diseñado por algún tipo de civilización no muy antigua.</p>

<p>Wrap Portal Verde</p> 	<p>Objeto interactuable. El jugador puede usarlo mediante el uso de la tecla Barra Espaciadora.</p>	<p>Al pulsar la tecla correspondiente, el jugador será teletransportado de un lado del portal hasta el otro.</p>	<p>Energía del cosmos condensada para permitir la comunicación entre dos puntos dimensionales separados.</p>
<p>Wrap Portal Rojo</p> 	<p>Objeto interactuable. El jugador puede usarlo mediante el uso de la tecla Barra Espaciadora.</p>	<p>Al pulsar la tecla correspondiente, el jugador será teletransportado de un lado del portal hasta el otro.</p>	<p>Energía del cosmos condensada para permitir la comunicación entre dos puntos dimensionales separados.</p>
<p>Elevator</p> 	<p>Objeto interactuable. El jugador reaccionará al elemento mediante el contacto con él.</p>	<p>Plataforma móvil de estilo decorativo. Podrá ser usada por el jugador, y deberá ser activada para su funcionamiento, el cual le permitirá moverse entre dos puntos.</p>	<p>Elemento tecnológico creado para facilitar la vida de alguna civilización antigua,</p>
<p>Plataforma Móvil 1</p> 	<p>Objeto interactuable. El jugador reaccionará al elemento mediante el contacto con él.</p>	<p>Plataforma que se mueve sistemáticamente entre dos puntos. Podrá ser usado por el protagonista. Podrá estar activado o desactivado, y podrá ser interactuado por otros elementos del nivel.</p>	<p>Elemento tecnológico creado para facilitar la vida de alguna civilización antigua,</p>
<p>Plataforma Móvil 2</p> 	<p>Objeto interactuable. El jugador reaccionará al elemento mediante el contacto con él.</p>	<p>Plataforma que se mueve sistemáticamente entre dos puntos. Podrá ser usado por el protagonista. Podrá estar activado o desactivado, y podrá ser interactuado por otros elementos del nivel.</p>	<p>Elemento tecnológico creado para facilitar la vida de alguna civilización antigua,</p>

<p>Plataforma Destrutable</p> 	<p>Objeto interactuable. El jugador reaccionará al elemento el uso de ataques contra él.</p>	<p>Plataforma que tiene la capacidad de desaparecer si sufre un número suficiente de ataques por parte del jugador.</p>	<p>Rocas y trozos de tierra colocados a propósito, con el fin de bloquear el acceso a los invasores.</p>
<p>Plataforma no rígida</p> 	<p>Objeto interactuable. El jugador reaccionará al elemento mediante el contacto con él.</p>	<p>Plataforma que se va disolviendo hasta desaparecer cuando contacta con el jugador. Reaparece después de un tiempo.</p>	<p>Terreno arcaico con propiedades mágicas, que permiten reacciones exóticas en la materia.</p>

Jugabilidad

Visión general del Juego

El juego se basa en los géneros plataformas y acción, aunque también bebe de aventura y puzzles, dado a que deberás explorar los distintos niveles y deberás realizar distintos rompecabezas que te otorgarán ciertas recompensas al resolverlos.

El juego será desarrollado para PC, será offline y de un solo jugador. Al jugar en solitario, se creará una experiencia más inmersiva en el juego, aumentando tu concentración y concentración durante el juego. Tendrá dos modos de juego, normal y difícil.

Experiencia del jugador

Vivirás la emoción e intriga de la supervivencia en un lugar recóndito y desconocido, donde podrás ir mejorando a tu personaje, superando así los distintos niveles y obteniendo nuevas herramientas y nuevas mejoras.

Deléitate de los hermosos paisajes de cada nivel. Corre, salta, asesina a tus enemigos y fortalécete mientras avanzas en la historia principal.

Pautas de jugabilidad

Siendo un juego donde deberás desenvolverte y matar monstruos, cierto nivel de dificultad y violencia es de esperarse. No obstante, ayudándonos del estilo de diseño y a la fantasía pretendemos alejarnos un poco del realismo y la brutalidad de los golpes, haciéndolo apto para un mayor rango de edades.

El juego se irá volviendo más difícil conforme se vaya avanzando en los niveles y la historia, lo que le irá aportando emoción y evitará la monotonía en la realización de cada nivel.

Se usará un lenguaje maduro con ciertos toques de ironía, aunque no será muy complejo. Se podrán encontrar referencias sutiles repartidas por los niveles, niveles que serán finitos.

Objetivos del juego y Recompensas

El jugador irá obteniendo nuevas herramientas y módulos conforme vaya recorriendo los niveles. El loot será aleatorio, pero habrá momentos donde recibas un objeto de forma asegurada, esto es, al derrotar a un boss y al realizar correctamente un puzzle entre niveles.

Como se explicará más tarde, las zonas o niveles son afectados por el ciclo de día y noche, y algunos fenómenos meteorológicos, los cuales ofrecerán de forma directa una ventaja o desventaja al jugador.

También podemos encontrar una desventaja que no parece muy evidente. Esta es tener mala suerte y que te poque poco loot, o que no resuelvas bien los puzzles. Esto supondrá, a la larga, tener menos opciones, y seguramente, menos capacidades para poder realizar los niveles correctamente.

La dificultad de cada nivel se irá incrementando, de forma que los enemigos tendrán más nivel y no dejarán tanto loot. Obviamente, los objetos dejados por estos monstruos más fuertes, serán de mejor calidad, por lo que conforme se avance hacia el final será mucho más importante la recolección de estos. Esto da por hecho que el jugador se acostumbrará al juego y se volverá más experto en sus mecánicas.

Mecánicas del juego

Atributos del personaje principal

El personaje es un humano medio cyborg, modificado con tecnología avanzada, con un brazo robótico que le permite, no solo tener una fuerza sobrehumana, sino también altos reflejos a la hora de combatir. Pero lo más importante es que el brazo tiene tres ranuras en las que, más adelante del juego, podrá insertar distintos tipos de *Módulos del ocaso* (ver más adelante). Puede equiparse también con armas de distintos tipos (espada corta, maza, lanza y cañón láser) y trajes que le proporcionarán habilidades pasivas. Tanto las armas como los trajes tendrán un peso que irá variando y afectará a la capacidad de movilidad del personaje al llevarlos equipados.

El personaje podrá realizar los movimientos básicos de un plataformas: saltar, moverse hacia los lados, hacia abajo y correr. Además podrá realizar acciones como atacar con el arma que esté portando y hacer cambios de arma, podrá realizar evasión para esquivar golpes enemigos y podrá usar módulos que tenga a su disposición. El personaje dispondrá de una vida y resistencia limitadas, además de atributos y estadísticas del personaje.

Las estadísticas del personaje principal son:

- **Nombre:** [*string*] Nombre que recibe el personaje protagonista.
- **Ataque:** [*float*] El ataque total que posee el personaje.
- **Defensa:** [*float*] La defensa total que posee el personaje.
- **Resistencia:** [*float*] La resistencia total que posee el personaje.
- **Salud:** [*float*] Los puntos de salud totales que posee el personaje.
- **Peso:** [*float*] El peso total que posee el personaje.
- **Arma:** [*arma*] El arma que tiene equipada actualmente el personaje.
- **Traje:** [*traje*] El traje que tiene equipado actualmente el personaje.
- **Módulos:** [*módulos*] Módulos del Ocaso que tiene conectados actualmente el personaje.

Además, el personaje tiene otros atributos ocultos que no son mostrados al usuario:

- **Vel. Ataque:** [*float*] La velocidad con la que ataca el personaje.
- **Vel. Movimiento:** [*float*] La velocidad con la que se mueve el personaje.
- **Evasión:** [*float*] El tiempo en segundos de invulnerabilidad que tiene el personaje al evadir.
- **Gravedad:** [*float*] La fuerza de gravedad con la que es atraído hacia abajo el personaje.
- **Salto:** [*float*] La altura con la que el personaje puede saltar.

Cálculo del daño recibido

El daño que va a recibir el personaje de los enemigos se calcula a partir de la siguiente fórmula:

$$dañoRecibido = 5 * \frac{atqEnemigo}{\sqrt{defPersonaje}} * (1 - dañoReducido)$$

siendo *atqEnemigo* el ataque del enemigo que provoca el daño al jugador (dependiente del nivel del enemigo), *defPersonaje* la defensa del personaje y *dañoReducido* la reducción del daño del personaje.

Atributos de los enemigos

Los enemigos son entidades que se encuentran repartidas por todas las zonas del mapa. Los hay de distintos tipos y de distintos niveles. Cuanto mayor sea el nivel del enemigo, mejores serán sus estadísticas.

Según el tipo de enemigo podrán moverse, saltar, lanzar proyectiles, e incluso evadir un ataque al protagonista.

Las características principales de los enemigos son:

- **ID:** [*int*] número de identificación del módulo dentro del juego.
- **Nivel:** [*int*] El nivel del enemigo.
- **Ataque:** [*float*] El ataque base que posee el enemigo.
- **Defensa:** [*float*] La defensa base que posee el enemigo.
- **Salud:** [*int*] Los puntos de salud base que posee el enemigo.
- **Nombre:** [*string*] nombre del enemigo.
- **Descripción:** [*string*] descripción del enemigo.
- **Vel. Ataque:** [*float*] La velocidad con la que ataca el enemigo.
- **Vel. Movimiento:** [*float*] La velocidad con la que se mueve el enemigo.

Todas las características físicas de los enemigos escalan con su nivel según la fórmula recursiva

$$var(n) = var(n - 1) \cdot \frac{(n-1) \cdot a}{2}, \quad \forall n \in [2, nivelMax]$$

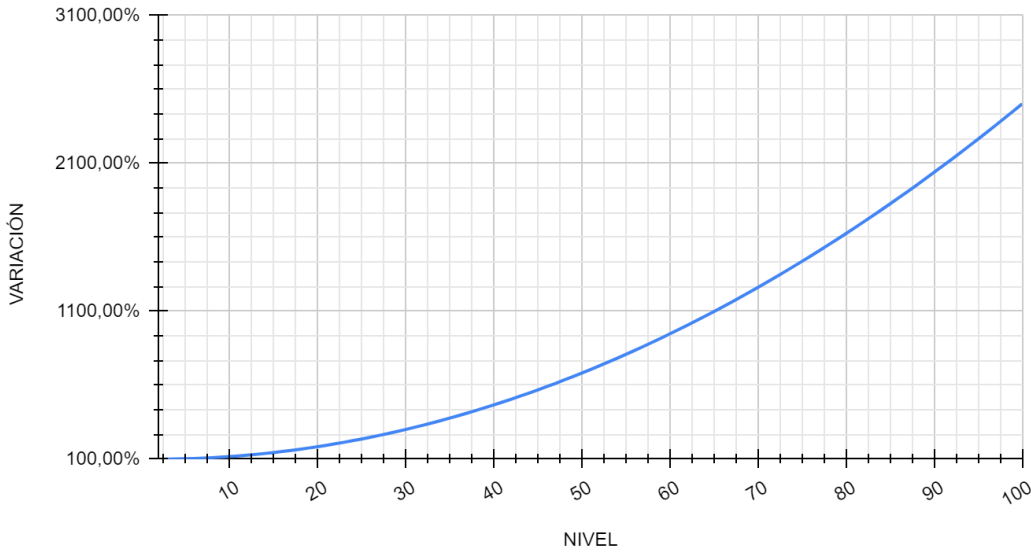
que, al convertir en forma iterativa, se queda como

$$var(n) = 100\% + \sum_{i=2}^n \frac{(i-1) \cdot a}{2} = 100\% + \frac{a}{2} \cdot \sum_{i=2}^n (i - 1) = 100\% + \frac{n(n-1) \cdot a}{4}$$

siendo n el nivel del enemigo y a la constante de aumento en el que se basa la fórmula. El porcentaje de variación que se obtiene se aplica sobre las características bases del enemigo. El nivel máximo que puede alcanzar un enemigo por sí solo (en la fórmula como $nivelMax$) es 100.


Para $a = 0,99\%$, se obtiene la siguiente gráfica:

VARIACIÓN GRÁFICA




Data: [Enemigos](#)

Items y Recursos

ID	ICONO	NOMBRE	RAREZA	DESCRIPCIÓN	USABLE
0		NONE	1	NONE	No
1		Degiterio	1	Recurso utilizado para canjear en el Nexo Central.	No

2		Occaterio	4	Material raro que se puede emplear para canjear en el Nexo Central, entre otras cosas.	No
3		Suero vital	2	Suero que restaura unos pocos puntos de salud.	Sí
4		Suero energético	2	Suero que reduce un poco el consumo de la resistencia.	Sí
5		Suero fortalecedor	2	Suero que aumenta ligeramente el ataque.	Sí
6		Suero protector	2	Suero que aumenta ligeramente la defensa.	Sí
7		Mineral fragmentado	2	Pedazo de mineral desbordado de las paredes de las cuevas de Oriquaht.	No
8		Nexotek	5	Cápsula que permite la unión con el Gran Nexo Central. Consúmelo para teletransportarte al último Nexo que te conectaste.	Sí
9		Mineral compacto	3	Trozo de mineral consistente formado por minerales más pequeños. Se puede también encontrar en ciertas zonas de las Cuevas del Alba.	No
10		Piedras de lava	3	Residuos de piedras pequeñas formadas por la lava de Oriquath. No se puede manipular directamente con las manos debido a su alta temperatura, pero sí al utilizar algún tipo de protección metálica que resista el calor.	No
11		Roca de magma	4	Roca bañada en un poderoso e incandescente magma creada por la compresión de pequeñas piedras de lava. Quema todo lo que se le acerque.	No
12		Sollozos del crepúsculo	4	Ciertas piedras de las que se desconoce el origen y la formación. Si las aprietas con fuerza, con suerte se pueden oír los sollozos de aquellas que serán víctimas de la Catástrofe del Principio y Fin.	No

13		Temor del crepúsculo	5	Gran roca que amacena un poderoso y siniestro poder. Al sujetarla con las manos, se puede sentir el miedo y los gritos de dolor de esos que perecerán el la Catástrofe del Principio y Fin.	No
----	---	----------------------	---	---	----

Data: [Items](#)

Modos de juego

1. Modo Normal

Modo de juego por defecto. La vista es en tercera persona. Este modo de juego es básicamente la dificultad para la que está pensada la historia y los objetivos, para que sea un juego entretenido e interesante. Está guiada hacia un público más general que quiera disfrutar de un juego con cierta dificultad y a la vez no sea frustrante por su dificultad, que reduciría así la apreciación del desarrollo de la historia.

2. Modo Difícil

Modo de juego opcional. La vista es en tercera persona. Este modo de juego está realizado para jugadores avanzados que quieran proponerse retos. Realmente no cambia mucho con el modo original, pero el simple hecho de un decremento de los atributos del personaje y aumento en los de los enemigos, junto a cambios en las probabilidades, supone un gran escalón de dificultad. Aquí, el jugador experimentará más intensidad, más tensión y más frustración, lo que dará una perspectiva más hostil e interesante a los jugadores veteranos.

Equipamiento - Sistema de puntuación del Personaje

El personaje principal no tiene sistema de niveles, por lo que cualquier aumento en su daño, vida y demás atributos dependen exclusivamente de su equipación.

ARMAS

Las armas son equipamiento que podrá utilizar el personaje para defenderse y atacar a los enemigos (excluyendo el brazo). Cada arma tiene las siguientes características:

- **ID:** *[int]* número de identificación del arma dentro del juego. No distingue entre los tipos de armas.
- **Tipo:** *[struct]* el tipo de arma. Puede ser *Espada Corta*, *Maza*, *Lanza* o *Cañón Láser*.
- **Ataque:** *[float]* define el daño que hace ese arma por sí sola.
- **Peso:** *[float]* define la masa de ese arma. Esta variable afecta al movimiento y desplazamiento del personaje, así como a la stamina que usará este al golpear con el arma.
- **Nombre:** *[string]* nombre del arma.
- **Descripción:** *[string]* descripción del arma.

ID	TIPO	ATAQUE	PESO	NOMBRE	DESCRIPCIÓN
0	Espada	0	0	Unnamed	Unnamed
1	Espada	3,0	4,0	Espada Corta	Una espada de corto alcance forjada para aquellos que poseen el coraje de enfrentarse a este mundo.
2	Maza	5,0	7,0	Maza	Una maza robusta pensada para demoler a las bestias que habitan las cavernas. Pesada pero poderosa.
3	Laser	1,0	2,0	Cañón Láser	Un cañón que canaliza energía para disparar pequeñas balas láser a gran velocidad.
4	Espada	3,5	4,3	Sable picapiedra	
5	Maza	6,8	8,2	Porra demolerocas	
6	Laser	1,9	3,3	Pistola rompemuros	

7	Espada	3,2	3,7		
8	Espada	5	6,1		
9	Laser	2,5	4		
10	Lanza	4,0	5,0	Lanza ligera	Una lanza de medio alcance utilizada por los guerreros más salvajes y feroces de este mundo.
11	Lanza	5,1	6	Lanza pesada	

Data: [Armas](#)

TRAJES

Los trajes son equipamiento que cubrirá al personaje con el propósito de reducir el daño recibido por el entorno y de otorgar habilidades pasivas que pueden, o bien activarse dado una condición o condiciones, o bien permanecer siempre activas. Cada traje tiene las siguientes características:

- **ID:** [*int*] número de identificación del traje dentro del juego.
- **Defensa:** [*float*] define la reducción del daño del traje por sí solo que recibirá el personaje.
- **Peso:** [*float*] define la masa de ese traje. Esta variable afecta al movimiento y desplazamiento del personaje.
- **Habilidad:** [*script*] define la habilidad pasiva del traje.
- **Nombre:** [*string*] nombre del traje.
- **Descripción:** [*string*] descripción del traje.

ID	DEFENSA	PESO	HABILIDAD	NOMBRE	DESCRIPCIÓN
250	1,0	2,0	Ninguna	Túnica protectora	Vestimenta que utilizó alguna civilización antigua para protegerse contra otras civilizaciones.
251	1,3	2,5	Aumenta los puntos de salud en 20	Traje de vitalidad	Un traje tejido con plantas místicas que cura las heridas de los guerreros más débiles.
252	1,6	3,5	Aumenta el ataque un 12%	Manto de la furia	El vestido que concentra la ira necesaria para luchar en combate sin miedo a la debilidad.
253	1,4	3,0	Aumenta la velocidad de ataque en un 15%	Ropa de los Rauvnir	Ropa que solía pertenecer a la raza de los Rauvnir, quienes tenían una brutal agilidad en combate con las

					armas.
254	1,4	3,0	Aumenta la velocidad de movimiento en un 15%	Túnica del mensajero	Los mensajeros usaban estas prendas para desplazarse rápidamente por las tierras de Oriquath, pero algunos cazadores y guerreros las usaban para huir de sus presas o perseguirlas.
255	1,7	3,5	Aumenta la velocidad de ataque en un 7% y la velocidad de movimiento en un 8% cuando la salud se encuentre por encima del 75%.	Túnica ligera	
256	2,2	4,0	Al derrotar un enemigo, el personaje recupera un tanto por ciento de salud equivalente al nivel del enemigo derrotado	Vestido del cosechador	
257	2,5	7,0	Cuando el personaje se encuentra en el aire, hay un 30% de probabilidad de no consumir resistencia al intentar consumirla.	Conjunto volador	
258	2,1	4,5	Cuando la salud del personaje se encuentra por debajo del 20%, el ataque aumenta un 50%	Armadura de Ftreqis	
259	2,0	5,0	Al activar un módulo del ocaso, los tiempos de espera de todos los módulos equipados se reducen en un 15%, pero la velocidad de movimiento se reduce en un 5%. Este efecto se puede acumular hasta 3 veces como máximo y cada efecto desaparecerá cuando se pueda volver a	Sábana del tiempo	

			usar cada Módulo del Ocaso.		
260	2,5	5,0	Aumenta la probabilidad de que haya tormenta	Bendición de las nubes	
261	1,8	4,0	Al llevar equipado una espada ligera, el daño inflingido aumentará en 10/15/25% según se tenga equipado 1/2/3 Módulos del Ocaso respectivamente.	Manto de la locura	Vestido legendario creado mediante artes oscuras que saca la demencia más profunda del alma.

Data: [Trajes](#)

MÓDULOS DEL OCASO

Los *Módulos del Ocaso* son unos módulos esféricos que se pueden insertar en el brazo robótico del personaje. Estos módulos proporcionan habilidades distintas a las de los trajes, mucho más poderosas y útiles. Algunos de los módulos del ocaso son tan poderosos que necesitan de varias esferas para poder funcionar. Las habilidades no son pasivas, es decir, las puede accionar el jugador cuando desee. Cada módulo del ocaso tiene las siguientes características:

- **ID:** [*int*] número de identificación del módulo dentro del juego.
- **Ranuras:** [*int*] indica las ranuras necesarias para que el módulo funcione. Estas ocupan las ranuras del brazo, que tiene 3 como máximo.
- **Duración:** [*float*] tiempo en segundos que dura el efecto de la habilidad del módulo.
- **TdE:** [*float*] tiempo de espera en segundos para poder volver a usar la habilidad del módulo una vez finalizado su efecto.
- **Habilidad:** [*script*] define la habilidad del módulo.
- **Nombre:** [*string*] nombre del módulo.
- **Descripción:** [*string*] descripción del módulo.

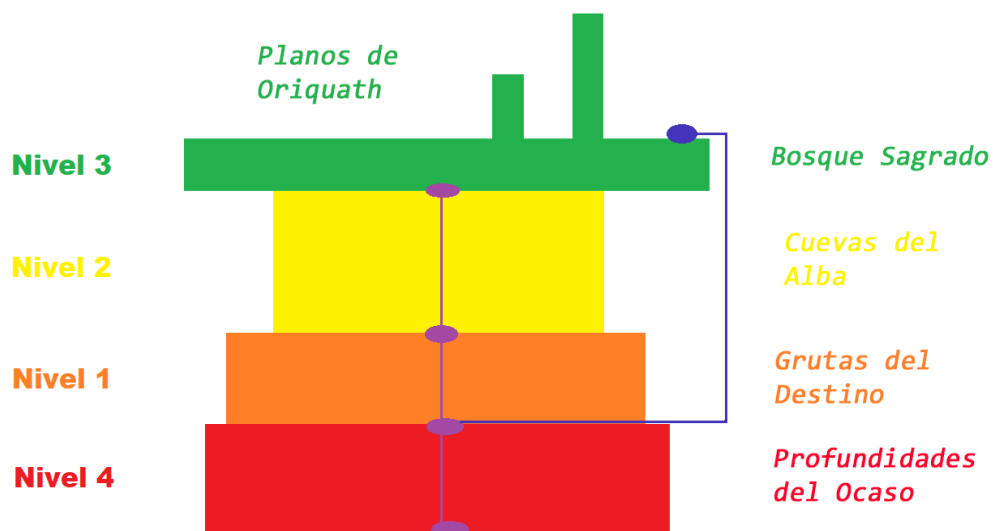
ID	RANURAS	DURACIÓN	TdEspera	HABILIDAD	NOMBRE	DESCRIPCIÓN
----	---------	----------	----------	-----------	--------	-------------

1	1	15,0	30,0	Durante el periodo del efecto, la capacidad de salto aumenta en un 50/75/100% .	Módulo de supersalto	Un sencillo módulo desarrollado y programado para los primeros prototipos de Neyteria. Al conectarse con estos, se puede llegar más alto al pegar un salto. Útil para llegar a zonas que de normal no se alcanzan con un simple salto.
2	2	5,0	60,0	Durante el periodo del efecto, los puntos de salud no decrecen.	Módulo de invencibilidad	Desde la Guerra del Ocaso se empezaron a desarrollar estos poderosos módulos, fuertes pero con limitaciones. En la guerra, muchas vidas se perdieron instantáneamente, con un simple toque de armas. Fue por eso que, después de dicha batalla, se desarrollo este módulo capaz de resistir cualquier tipo de ataque durante un corto periodo de tiempo.
3	2	7,0	60,0	Durante el periodo del efecto, la resistencia no decrece.	Módulo de vigor	Aquellos ágiles valientes que pudieron esquivar los ataques de sus rivales y contraatacarlos a la vez, perecieron al poco tiempo de lucha por falta de resistencia. Pues para ello se desarrolló este módulo, que permite extender sorprendentemente el aguante de su portador.
4	1	10,0	25,0	Durante el periodo de efecto, el daño recibido se verá reducido en un 20/40/60%	Módulo de Rumarh	Se cuenta que este Módulo del Ocaso se diseñó con la idea de asemejarse a uno de los guerreros gigantes más resistentes. Se llamaba Rumarh y era admirado por su capacidad de resistir los golpes con mayor facilidad.

Data: [Módulos](#)

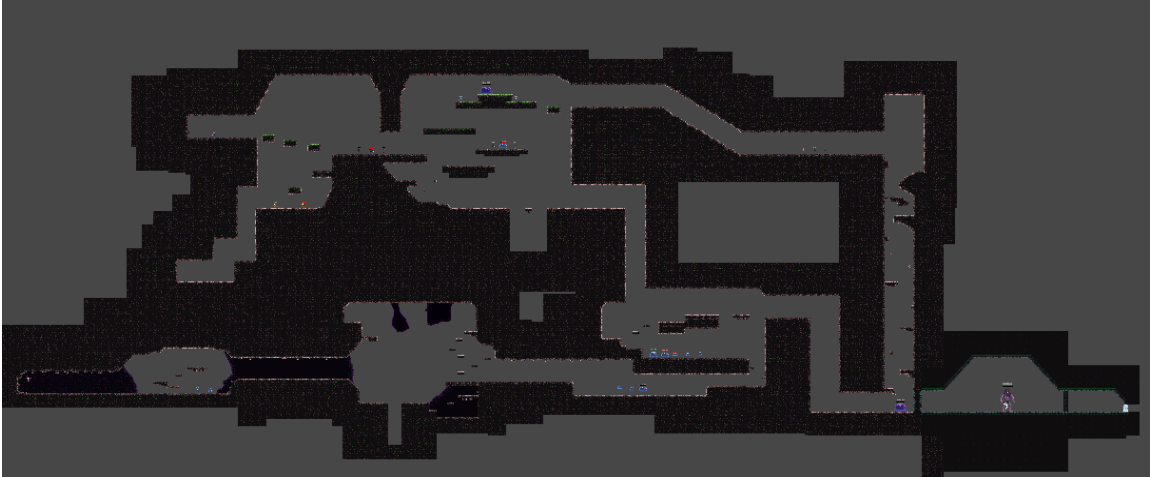
Diseño de nivel. Mecánicas

El mapa del juego está distribuido por zonas o niveles, que a su vez están agrupadas en distintas regiones. Todas las zonas están conectadas en un mismo espacio, es decir, que realmente no hay pantallas de carga de cara al usuario, sino que se van cargando las zonas adyacentes a la que se encuentra el personaje (totalmente sujeto a cambios en caso de que no se pudiese implementar). El mapa tendrá un total de 4 niveles, con temáticas, paisajes y ambientes diferentes, cada uno con un boss final de zona. Se dispondrá, además, de un notable incremento de la dificultad conforme vayamos superando cada nivel. (Falta la explicación de cada nivel, sobre todo artísticamente).

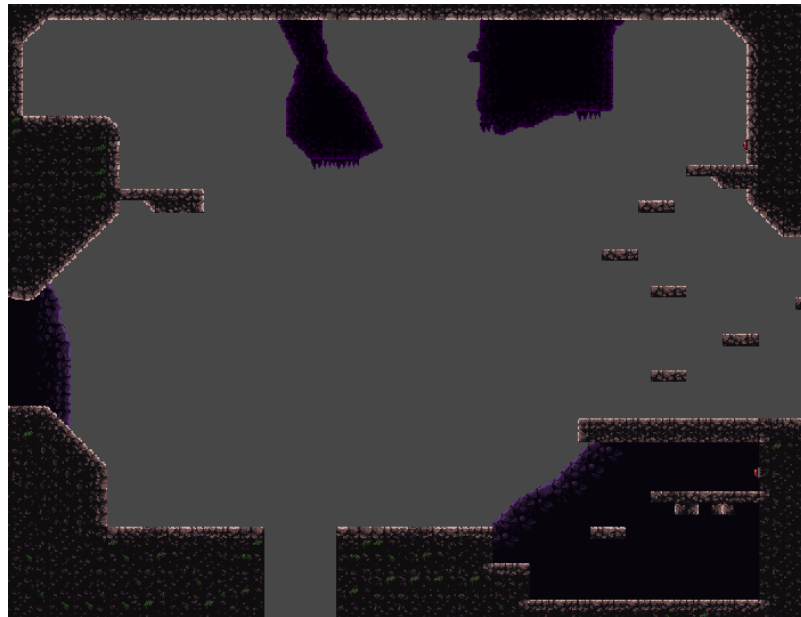


1. Nivel 1 - Grutas del Destino

Típica *estética de cuevas* a las que estamos acostumbrados. Un poco de oscuridad, tonos un poco oscuros y enemigos de acuerdo a la temática.



Nivel 1 - imagen 1 (general)



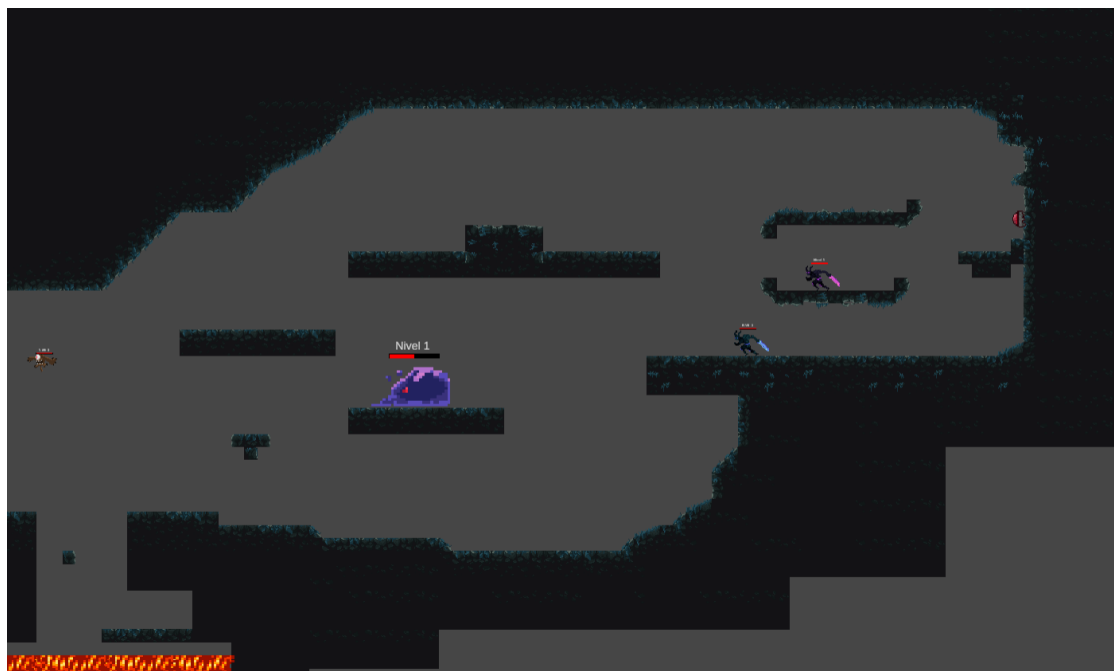
Nivel 1- imagen 2 (ejemplo de mapa desde cerca)

2. Nivel 2 - Cuevas del Alba

Estética de cuevas, pero con más **iluminación** y cierta **vegetación**. Tonos un poco más claros y enemigos de acuerdo a la temática.



Nivel 2 - imagen 1 (general)



Nivel 2 - imagen 2 (ejemplo de mapa desde cerca)

3. Nivel 3.1 - Planos de Oriquath

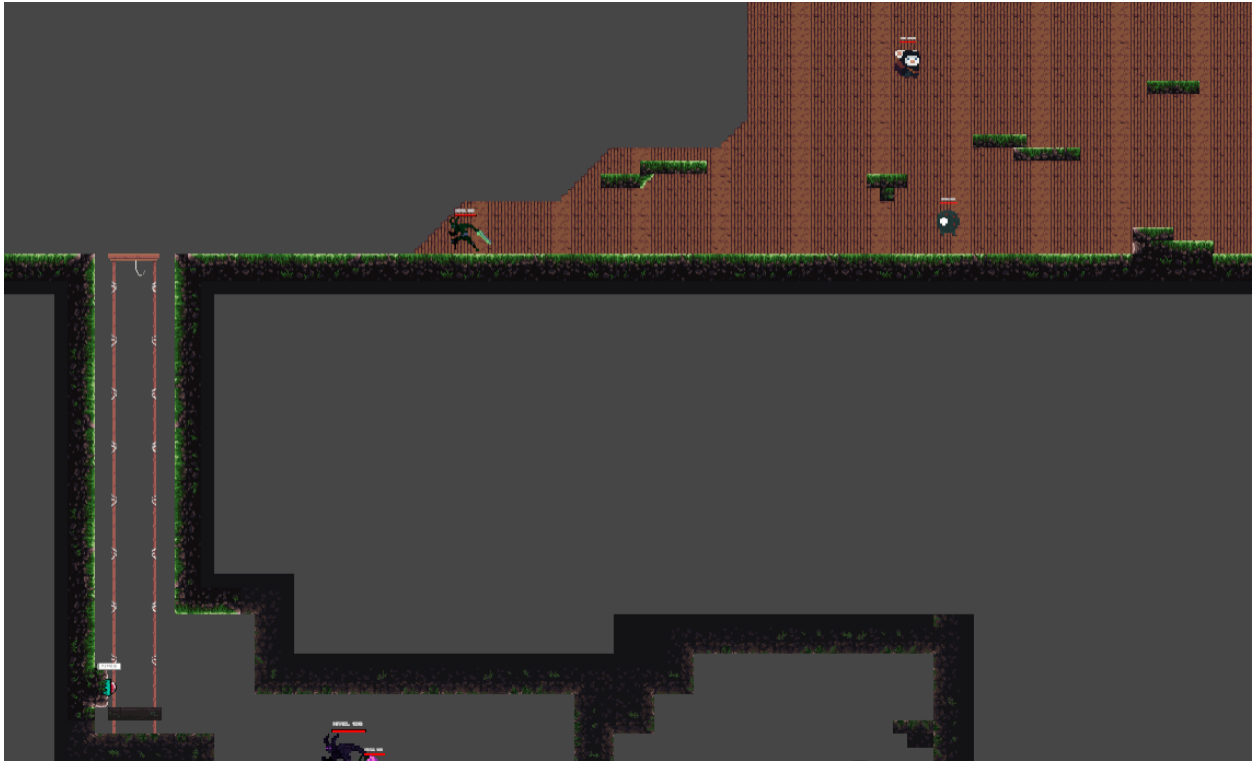
Estética de superficie con **vegetación** y edificios/infraestructuras como ruinas. Buena **iluminación** y fondo de mapa muy vistoso viendo los soles y alguna estrella. Enemigos de acuerdo a la temática.

Nivel 3.2 - Bosque Sagrado

Estética de superficie con más **vegetación** y pocas ruinas. Buena **iluminación** y fondo de mapa muy vistoso con buena vista del cielo y los árboles. Enemigos de acuerdo a la temática.



Nivel 3 - imagen 1 (general)



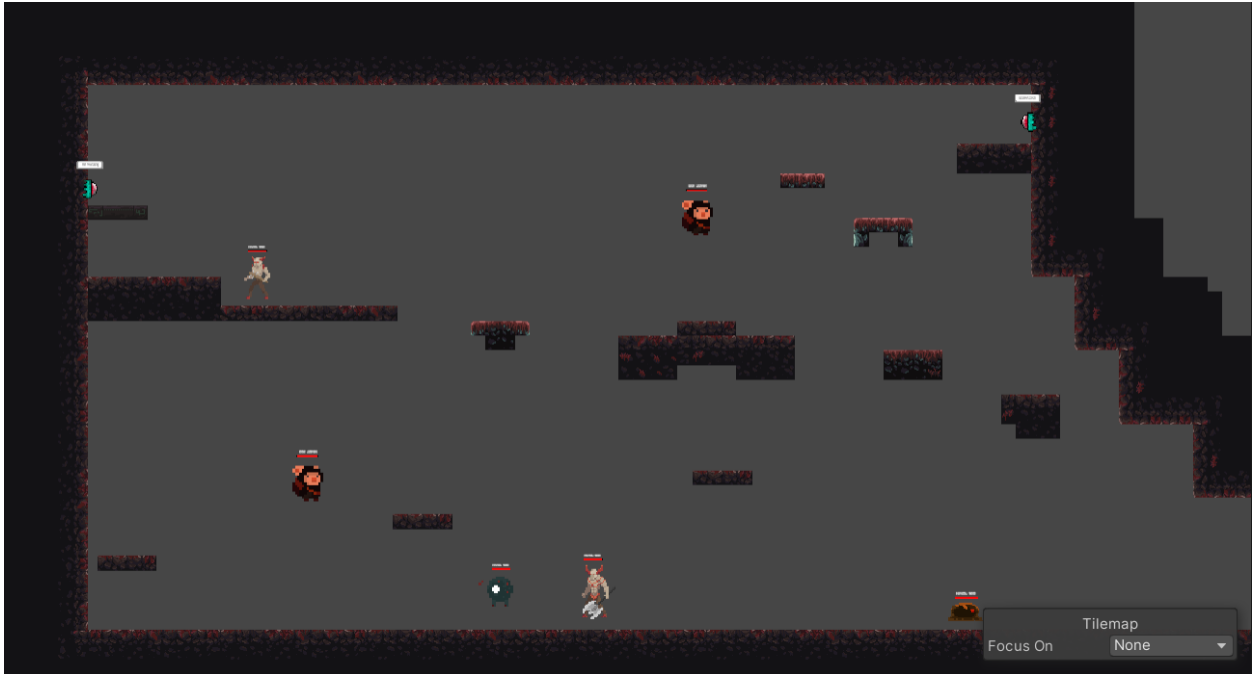
Nivel 3 - imagen 2 (ejemplo de mapa desde cerca)

4. Nivel 4 - Profundidades del Ocaso

Estética de cuevas, con más **oscuridad** y temática muy árida. Tonos más **rojizos**, vegetación nula, y decoración de **lava** abundante. Enemigos de acuerdo a la temática.



Nivel 4 - imagen 1 (general)



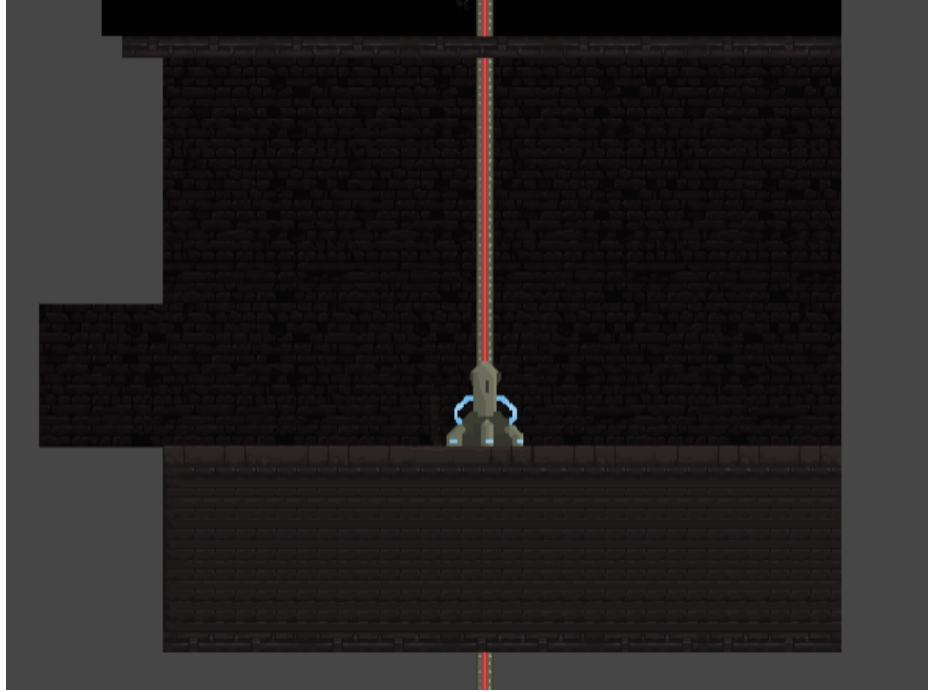
Nivel 4 - imagen 2 (ejemplo de mapa desde cerca)

5. RestZone - Zona de descanso entre niveles

Estética de edificio, con poca **oscuridad** y temática suave. Tonos marrones, vegetación nula, y decoración de infraestructura.



ZoneLoader - imagen 1



ZoneLoader - imagen 2

Mapa final con todos los niveles



Neyteria- imagen del mapa completo

CLIMA

Durante todo el tiempo, en cualquier zona del mapa, incluidas las subterráneas e interiores, el ciclo de día y noche y ciertos fenómenos meteorológicos pueden afectar considerablemente al entorno. En estas situaciones, por si se diera el caso de que el jugador no se encontrase al aire libre, se le notificará por pantalla que el clima ha cambiado. De igual manera, se le indicará cuando ha vuelto a la normalidad.

1. Día y Noche

Cuando cae la noche, los enemigos se fortalecen, incrementando su nivel en 5 unidades. En esta situación, el nivel de los enemigos se mostrará de color **naranja** y todos sus estados se verán escalados por el nivel. Además, el intervalo de invulnerabilidad de la evasión del personaje se verá reducido en un 25%.

Una vez termina la noche y amanece, el nivel vuelve al original.

2. **Tromenta**

El clima es muy inestable en este planeta, pues cuando llueve siempre hay tormenta. Es el mejor momento para enfrentarse a los enemigos, ya que la lluvia tiene ciertas propiedades que los debilitan, de modo que el nivel de estos se decrementa en 5 unidades. En esta situación, el nivel de los enemigos se mostrará de color **azul celeste** y todos sus estados se verán escalados por el nivel. También puede provocar que el personaje se resbale al desplazarse por la superficie. Además, se reduce el tiempo de espera de los Módulos del ocaso en un 25%.

Una vez termina la tormenta y el cielo se despeja, el nivel vuelve al original. La probabilidad de que haya tormenta es del 10% cada día (únicamente durante el día y en la superficie).

3. **Eclipse**

Es difícil que unos de los soles se cruce con una de las lunas, pero cuando esto sucede, el estado de la atmósfera se altera negativamente y las partículas que normalmente debilitan a los enemigos, invierten su efecto incluso a mayor escala. En esta situación, el nivel de los enemigos se incrementa en 6/7/8/9/10 unidades y se mostrará de color **rojo**, por lo que todos sus estados se verán escalados por el nivel. También afecta a la velocidad de movimiento del personaje, que se verá reducida en un 25%. Además, el tiempo de espera de los Módulos del ocaso se incrementará en un 25%.

Una vez finaliza el eclipse, el nivel vuelve al original. La probabilidad de que se produzca un eclipse es del 7/6/5/4/3% cada día, respectivamente (únicamente durante el día).

DROPS

NIVEL 1 - GRUTAS DEL DESTINO

El juego comienza en el nivel 1, que corresponde a la zona de las Grutas del Destino. El personaje se despierta en un pasillo oscuro repleto de partes robóticas averiadas.

Al salir del pasillo se llega a una pequeña zona de introducción con dos plataformas, una más alta que la otra, donde se le deja al jugador experimentar con el salto. En la plataforma más alta encontramos el primer Suministrador [1].

ENTIDAD	ITEM	CANTIDAD	RATE
Suministrador [1]	Degiterio	10	100%
Suministrador [2]	Degiterio	10	100%
	Occaterio	1	5%
	Mineral fragmentado	2	25%
Suministrador [3]	Degiterio	10	100%
	Occaterio	2	5%
	Suero vital	2	100%
Suministrador [4]	Degiterio	10	100%
	Occaterio	2 (A)	10%
	Mineral fragmentado	5 (A)	25%
Suministrador [5]	Degiterio	15	100%
	Occaterio	2 (A)	10%
	Módulo de supersalto	1	100%
Suministrador [6]	Degiterio	20 (A)	80%
	Occaterio	4 (A)	25%
	Mineral fragmentado	5 (A)	25%
	Suero energético	5 (A)	50%
Suministrador [7]	Degiterio	15	100%
	Occaterio	2	100%
	Suero vital	2 (A)	50%
	Suero energético	2 (A)	50%
Suministrador [8]	Sable picapiedra	1	100%
	Porra demoleroocas	1	100%
	Pistola rompemuros	1	100%
	Degiterio	30	100%
	Occaterio	5	100%
	Mineral fragmentado	5	100%

Sonidos del Juego

Los sonidos del juego se pueden categorizar en dos tipos, los relacionados con el ambiente y los que tienen que ver con la interacción con el juego.

En la parte ambiental podemos encontrar sonidos y efectos del nivel como aire, fauna y música de fondo de ambiente del nivel. Cada nivel tendrá su propia música, además de tener modificaciones distintas cuando sea día o noche y según la meteorología, representando una variación la intensidad o dificultad del nivel. También podemos encontrar música ambiental en el menú principal, un poco más activa para que llame al jugador a empezar a jugar.

En la parte de interacción podemos encontrar los ruidos de queja del personaje principal, el sonido de muerte, sonido de acción de las herramientas, las habilidades y los módulos; los sonidos de daño y muerte de cada enemigo y también los sonidos de interacción en el menú principal (al moverse y al pulsar por el menú principal y sus submenús).

Esquema de los controles

Como el juego está pensado para PC, su esquema de controles será para un teclado, aunque si se detecta un joystick se podrá utilizar como controlador también.

Tecla	Botón (Joystick)	Acción que ejecuta
W		El personaje realiza un salto.
A		Mueve al personaje hacia la izquierda.
D		Mueve al personaje hacia la derecha.
Shift		El personaje realiza una evasión.
Q		El personaje cambia de arma.
Click izquierdo		El personaje realiza un ataque con el arma equipada.
Click derecho		El personaje realiza un ataque especial con el arma equipada.
Espacio		El personaje interactúa con el entorno.
1		El personaje activa el módulo 1
2		El personaje activa el módulo 2.

3		El personaje activa el módulo 3.
E		El personaje utiliza el objeto rápido
←		El personaje cambia de objeto rápido (siguiente).
→		El personaje cambia de objeto rápido (anterior).
Ctrl + Click Izquierdo		El personaje fija la vista en un enemigo.
Rueda del ratón		El personaje cambia la vista fija entre enemigos.
Escape		Se pausa el juego y se muestran todos los menús.
F		Se muestra el inventario
R		Se muestran las estadísticas.
Z		Se muestra el mapa.

Estética del juego e interfaz de usuario

El juego va a tener una estética principalmente pixel art, con gamas de colores variables según el nivel correspondiente. Dada nuestra inexperiencia en el sector artístico intentaremos no cometer errores artísticos, pero no tenemos duda de que se nos escapan cosas (sobre todo en la iluminación).

El HUD y la interfaz del juego se explicarán a continuación:

HUD:

El HUD se colocará en la parte superior izquierda de la pantalla, dejando el resto del espacio disponible para que el jugador pueda disfrutar mejor el juego. Aunque, para menor visibilidad la barra de vida del jefe (aparte de estar encima de la entidad como los demás monstruos) estará en la parte inferior central de la pantalla, con su nombre. Cuando una entidad reciba daño o curación saldrá un texto indicando el daño producido o curado. También habrá un texto abajo a la izquierda semitransparente que indicará cuál es la zona/el nivel actual.

- La vida se representará mediante una barra verde.

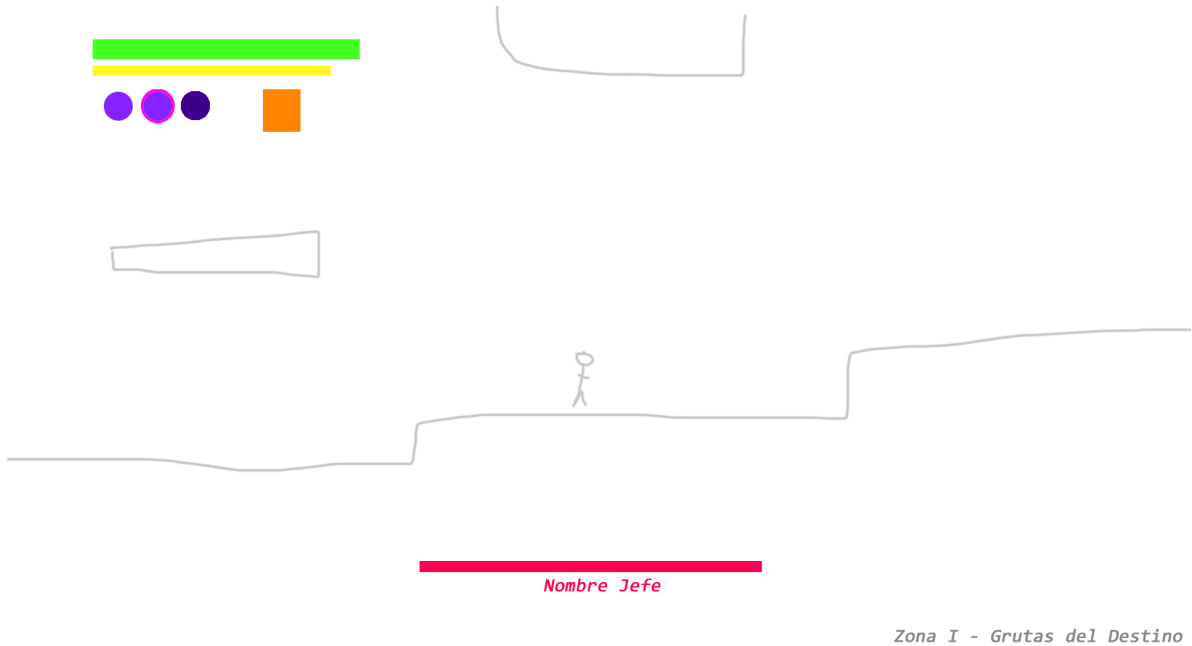
- La stamina mediante una barra amarilla debajo de la de vida. Esta barra se volverá azul unos instantes cuando se use una habilidad.
- Inmediatamente abajo se encontrarán tres círculos morados que representarán las ranuras para módulos del brazo, que indicará qué ranuras están disponibles y/o cuánto tiempo de espera hay hasta el siguiente uso del módulo correspondiente.
- A su derecha se encuentra un cuadrado naranja que indicará el ítem activable que está en ese momento preparado para su uso.



Hud del personaje detallado



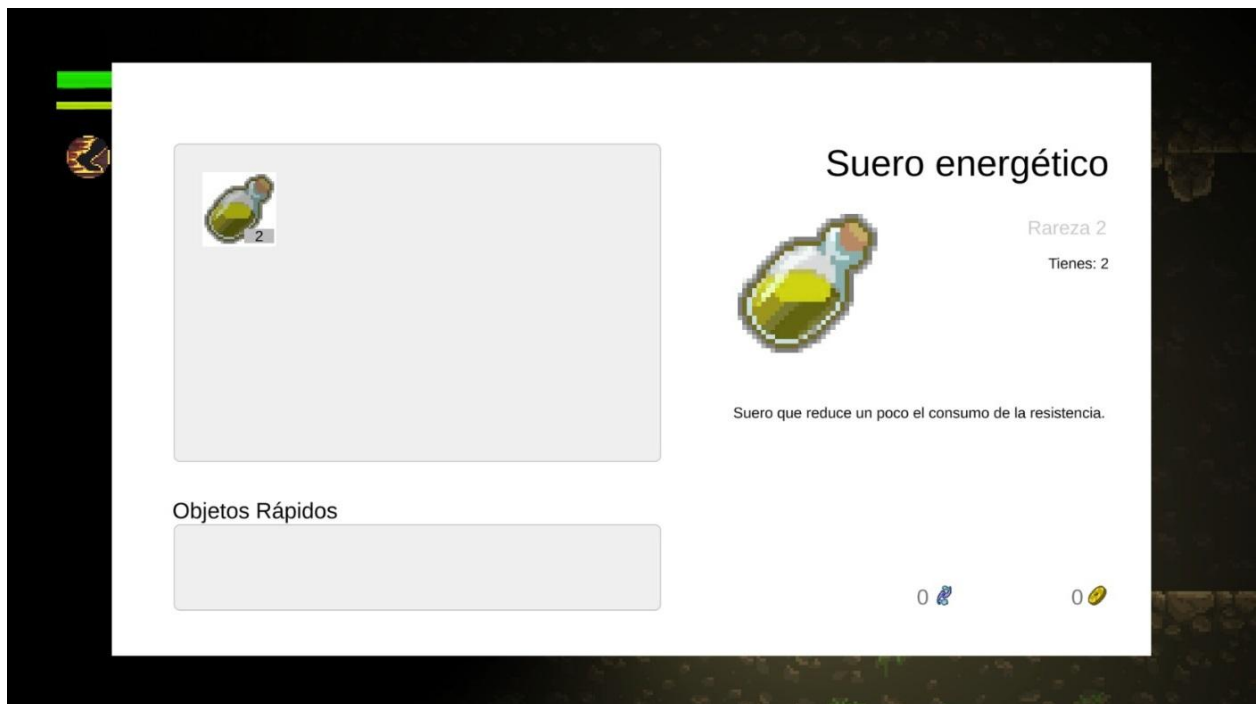
Visión de la cámara principal en el juego.



HUD y Visión de la cámara principal en el juego (Boceto).

Interfaz inventario:

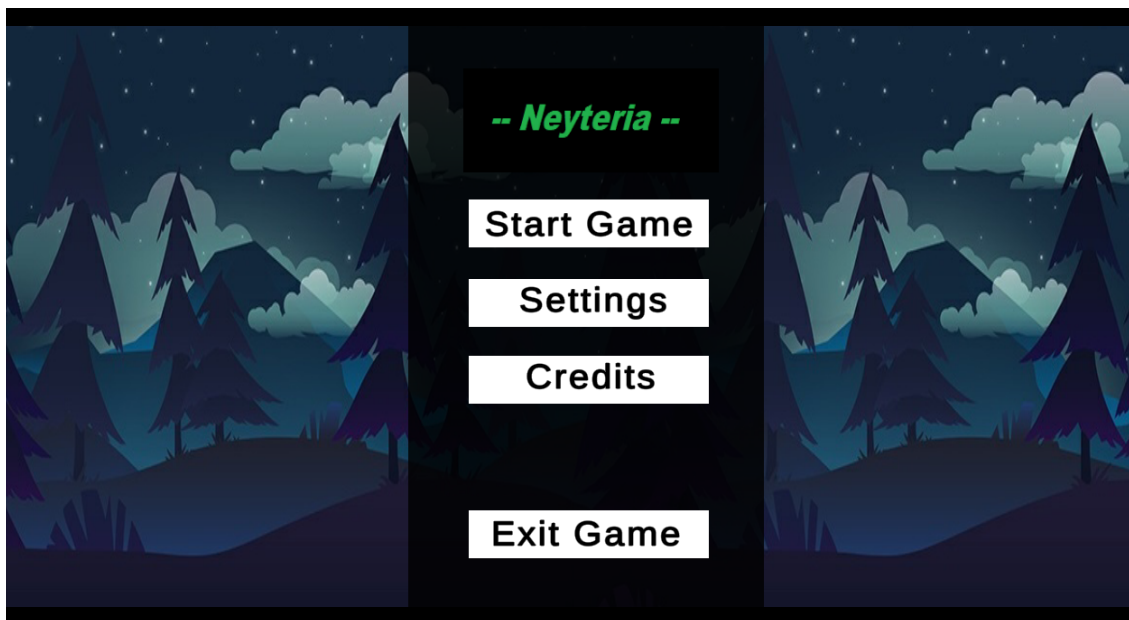
El inventario aparecerá en el centro de la pantalla, dejando el resto del espacio disponible ofreciendo una sensación más inmersiva en el juego.



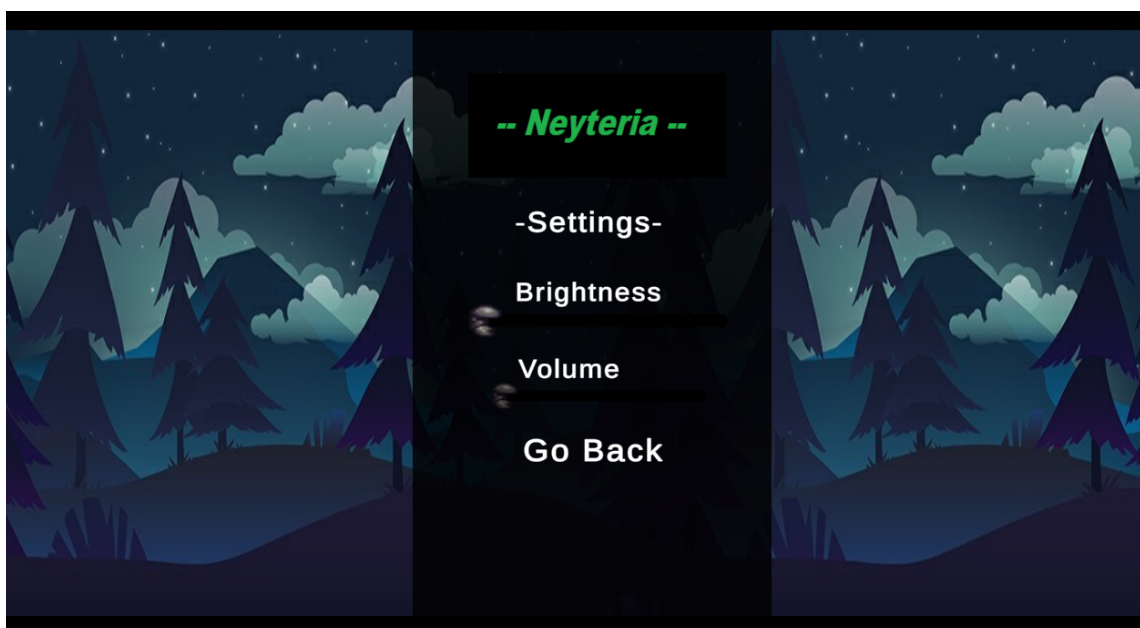
Vista del inventario en el juego.

Navegación del menú principal:

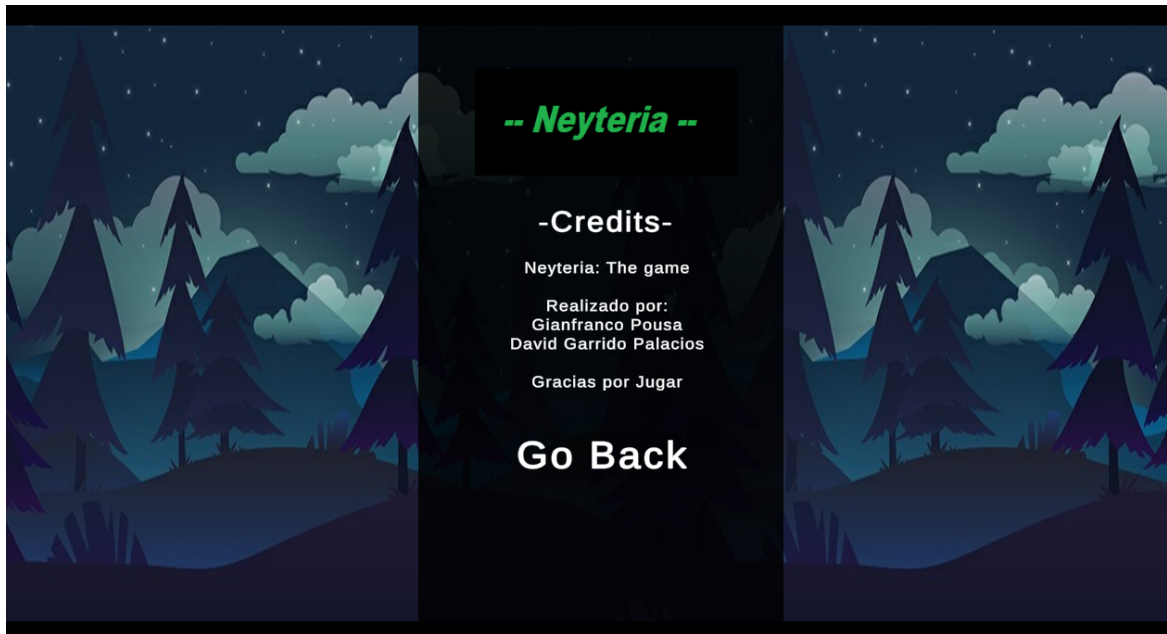
La Página Principal del juego dará acceso a la página de Start Game, Settings o Crédits. También se podrá salir del juego desde esta. Desde las demás páginas se podrá volver a la principal.



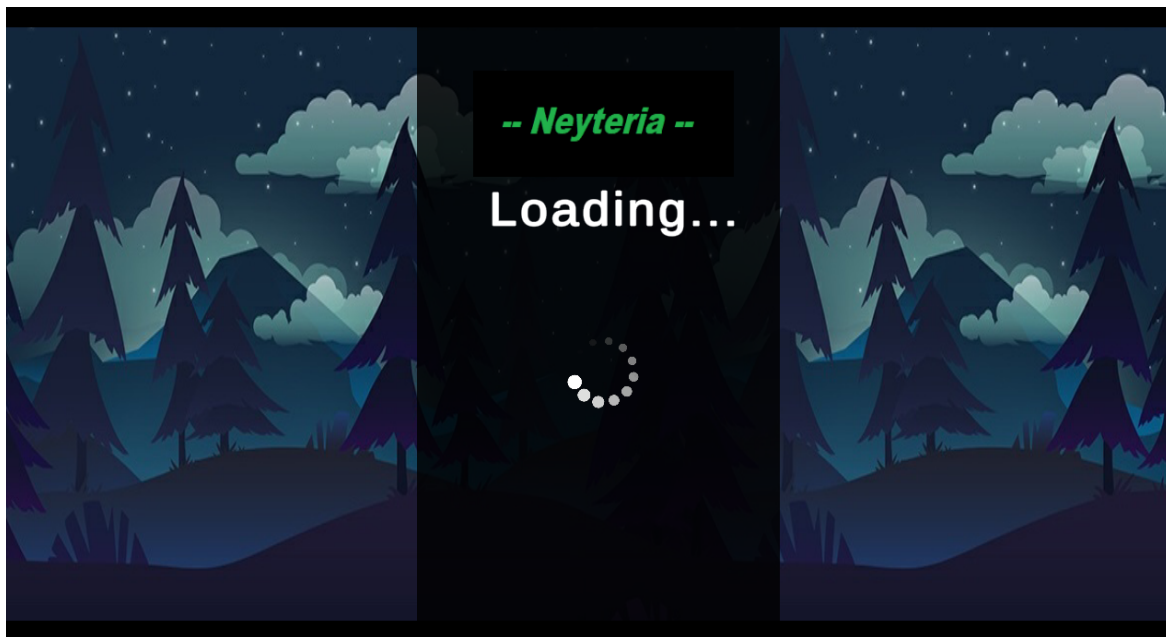
Página principal del juego.



Página de las configuraciones del juego "Settings".



Página de los créditos del juego "Credits".



Página de carga del juego "Start Game".

Pantalla de muerte:

La pantalla de muerte terminará el juego y dará acceso a la Principal.



Pantalla de muerte del juego "Game Over".

Comandos de la consola admin

Para abrir la consola en modo administrador, se debe pulsar las teclas “T + Enter” y para salir de la consola, basta con pulsar “Enter” sobre la línea de comandos. Al escribir un comando que contenga como mínimo los dos primeros términos, si se pulsa la tecla “Tabulador” aparecerá una ayuda en rojo mostrando los parámetros del comando que faltan por introducir.

COMANDO	PARÁMETROS	DESCRIPCIÓN
comm health <cant>	cant: <ul style="list-style-type: none"> - “max”: establece la salud a máximo. - [float]: otorga dichos puntos de salud. 	Comando para modificar la cantidad de salud del personaje principal.
comm selfkill		Comando para matar al personaje principal
comm player <atributo> <valor>	flag: <ul style="list-style-type: none"> - “health”. - “max_hp”. - “stamina”. - “max_stamina”. - “attack”. - “defense”. - “weight”. - “mov_speed”. - “jump_cap”. - “dash_range”. - “gasto_dash”. - “stamina_vel_rec”. - “dmg_reduc”. - “inmune”. - “stamina_decrease”. - “no_cd”. - “dash_on_air”. - “one_hit_kill”. 	Comando para modificar los atributos principales del personaje principal.
comm item <opt> <id> [cant]	opt: <ul style="list-style-type: none"> - “give”. - “remove”. 	Comando para otorgar o quitar objetos al personaje

	id: id del objeto cant: cantidad del objeto. Si no se especifica: - "give" -> otorga x1 - "remove" -> quita todos	principal.
--	--	------------

Tareas y Calendario

List the tasks that need to be completed along with the basic timeline to complete them by. The task list can be as detailed as you like to fit your studio's needs. The table below can be substituted for the excel file. This table is a great start but the tasks should be much more detailed.

Tasks to Complete & Schedule				
Tasks	Task Lead	Start	End	% Complete
Development Phase				
Design				
Storyline	Gianfranco	05/01/22	x/x/xx	80 %
Level Mechanics	David	05/01/22	x/x/xx	20 %
Art				
Level 1	Gianfranco	05/01/22	x/x/xx	5 %
Special FX				
UI	David	05/01/22	x/x/xx	10 %
Engineering				
Production Pipeline				
Prototypes	Gianfranco	05/01/22	x/x/xx	5 %
Audio				
Sound Design	David	09/01/22	x/x/xx	0 %
Milestone: GamePlay Features & Music				
Testing Phase				
Test Plan				
Beta Testing				
Milestone: QA Testing				
Deploying Phase				
"Go Live" Plans				
Milestone: Ready for Usage				